Ollscoil na hÉireann, Corcaigh

National University of Ireland, Cork

Resource Management and IP Interoperability for Low Power Wide Area Networks

Thesis presented by

Khaled Q. Abdelfadeel Hassan

MSc.

for the degree of

Doctor of Philosophy

University College Cork

Computer Science and Information Technology

Head of School: Prof. Cormac Sreenan

Supervisor(s): Prof. Dirk Pesch

2020

Research supported by Science Foundation Ireland

Contents

List of Figures

List of Tables

List of Abbreviations

3GPP 3rd Generation Partnership Project. 9, 10

6LoWPAN IPv6 over Low-Power Wireless Personal Area Networks. 13, 15, 35, 36, 38, 51, 140

6lo IPv6 over Networks of Resource-constrained Nodes. 35, 36

ACM Association for Computing Machinery. xv, 15, 140

ADR Adaptive Data Rate. 26, 30, 31, 33, 64, 67

BLE Bluetooth Low Energy. i, 3, 4, 35

BW BandWidth. 64, 68, 69, 75, 148–150

C/D Compression/Decompression. 40–42, 45–48, 54, 56, 61

CBOR Concise Binary Object Representation. 51

CIR Co-channel Interference Rejection. vi, 28, 63, 71, 81, 99–101

CoAP Constrained Application Protocol. i, 14, 35, 36, 39, 41, 42, 45, 48–50, 59, 140

CR Coding Rate. 64, 68, 69, 148–150

CRC Cyclic redundancy check. 26

CSMA Carrier Sense Multiple Access. 146

CSS Chirp Spread Spectrum. i, 7, 24

DER Data Extraction Rate. 14, 16, 65, 66, 70, 71, 77–82, 108, 110–112

DODAG Destination Oriented Direct Acyclic Graph. 51

DSSS Direct-Sequence Spread Spectrum. 24

eDRx Extended Discontinuous Reception. 9

FUOTA Firmware Updates Over The Air. iii, iv, vi, xiv, 10–12, 14, 15, 17, 18, 33, 34, 116–123, 125, 138, 143, 145, 146

GSM Global System for Mobile Communications. 10

HTTP HyperText Transfer Protocol. 49

ICMP Internet Control Message Protocol. 37, 41, 51, 52

IEEE Institute of Electrical and Electronics Engineering. xv, 2, 15, 140

IETF Internet Engineering Task Force. 13, 15, 36, 140

IID Interface IDentifier. 37

IoT Internet of Things. i, xiii, 1–9, 11, 13–15, 17, 24, 29, 35, 36, 44, 63, 144

IP Internet Protocol. 11, 21

LP-WiFi Low Power WiFi. i, 4

LPWAN Low Power Wide Area Network. i, iv, xiii, 5–16, 18–63, 139–141, 143–146

LSCHC Layered SCHC. i, iv, xvi, 36, 43, 44, 50–54, 62, 140

LTE Long Term Evolution. 9, 10, 19

MAC Medium Access Control. xiii, xiv, xvii, 2, 3, 7, 8, 13–18, 21–24, 30, 33, 37, 52, 63, 64, 84, 85, 87, 118, 139, 141, 142, 146, 148

MIC Message Integrity Code. 30

NB-IoT Narrow Band IoT. 9–11, 19

NFC Near Field Communication. 35

OSI Open Systems Interconnection. 35

PSM Power Saving Mode. 9

QoS Quality of Service. 5, 11, 12, 63

RFC Request For Comments. 37, 38

RFID Radio Frequency IDentification. 1

ROHC RObust Header Compression. 38, 39, 41

RPL Routing Protocol for Lossy Networks. 51

RRC Radio Resource Control. 10

RSSI Received Signal Strength Indicator. 21, 30, 63, 64, 67, 69, 71, 73, 75–83, 96

RTP Real Time Protocol. 38, 39

SCHC Static Context Header Compression. i, iv, vi, xiii, 15, 16, 18, 36, 39–56, 62, 140, 141, 144, 145

SDOs Standards Developing Organizations. 2

SF Spreading Factor. 53, 64, 66–71, 74, 75, 77–82, 148–150

SIG Special Interest Group. 3

SNR Signal Noise Ratio. 21, 30

SoC System on Chip. 2

SRB Signaling Radio Bearer. 10

TCP Transmission Control Protocol. 21, 37–39

TP Transmission Power. 66, 67, 70, 71, 73–77, 79, 82

TSCH Time Slotted Channel Hopping. 3

TSMP Time Synchronized Mesh Protocol. 3

TTL Time To Live. 46–48, 50, 55

UDP User Datagram Protocol. vi, 35–39, 41–44, 48–50, 52, 55, 59, 140

UNB Ultra Narrow Band. 7

WSN Wireless Sensor Network. 139

WSNs Wireless Sensor Networks. xiii, 7, 35, 139

I, Khaled Q. Abdelfadeel Hassan, certify that this thesis is my own work and has not been submitted for another degree at University College Cork or elsewhere.

Khaled Q. Abdelfadeel Hassan

To my parents

Acknowledgements

I would like heartily to express my appreciation and gratitude to Dirk Pesch for his supervision and support throughout my PhD research. I also extend my gratitude to Victor Cionca for his mentoring during my time at CIT. Additionally, I want to thank all my colleagues from CIT, UCC, Tyndall, and Danalto for being part of my PhD journey. Last but not least, I send my regards to my family and friends for their continued encouragement.

This thesis has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) and is co-funded under the European Regional Development Fund under Grant Number 13/RC/2077.

Khaled Q. Abdelfadeel Hassan
2020

Abstract

Low Power Wide Area Networks (LPWANs) such as LoRaWAN, Sigfox, and NB-IoT present a novel communication paradigm, which complements the traditional short-range Wireless Sensor Networks (WSNs). LPWANs promise to provide *wide-area* connectivity (up to tens of kilometers) and *low-power* operations (up to 10 years) for a *massive* number of *low-cost* devices. These unique features empower LPWANs to address diverse requirements of the Internet of Things (IoT) applications, from agriculture to smart cities.

Since the early days of LPWANs, a major hype has surrounded them, making it sometimes difficult to clearly understand and assess their capabilities. This has been inflated with biased reports and inaccurate data to promote even inapplicable solutions. Therefore, the thesis's goal has been set to assess the network stack of the LPWAN technologies, especially LoRaWAN, in different scenarios in order to understand their real advantages for the IoT applications and also to point out their drawbacks. Consequently, proposing improvements in order to enhance the performance and to extend the application domains of the LPWANs, taking into account their limitations, e.g., duty cycle. Specifically, resource management and IP interoperability topics are the main focus of the research. Within these topics, multiple novel contributions are made, which target different layers of the network stack, from the Medium Access Control (MAC) layer to the application layer.

The performance of the new Static Context Header Compression (SCHC) protocol was evaluated. As a result, two novel enhancements for SCHC are proposed in order to reduce its memory footprint and improve its compression efficiency. The proposed work improves the performance of SCHC in order to extend the Internet architecture to the LPWANs and thus enabling end-to-end IP connectivity and interoperability. Enabling end-to-end IP connectivity in LPWANs has the potential to bring the power of openness, interconnection, cooperation, and standards to their applications and devices.

The performance of the PYH and the MAC layers of LoRaWAN were evaluated, including its adaptive data rate mechanism. This study revealed an unfair data extraction rate among devices, favouring the close ones to the gateway and the ones that use high data rates. Consequently, a novel resource allocation mechanism, *FADR*, is proposed to enhance the fairness in LoRaWAN by managing the devices' data rates and transmission power levels. As a result, *FADR* achieves an

almost uniform data extraction rate for all devices regardless of their positions from the gateway and their data rates used.

LoRaWAN scalability and agility were studied for data collection applications wherein mobile gateways may participate in the collection processes. The results showed the poor performance of LoRaWAN in dense deployments and in confirmed traffic due to the severe collisions and the duty cycle limitation. To address this, a novel mechanism, called *FREE*, consists of a time-slotted MAC protocol and a resource management algorithm, was proposed on top of LoRaWAN. *FREE* manages the resources of the device (i.e. spreading factor, transmission power, frequency channels, etc.) and schedules the transmissions in time slots. As a consequence, *FREE* overcomes the poor scalability of LoRaWAN and can enable reliable and energy-efficient data collections.

The possibility of supporting Firmware Updates Over The Air (FUOTA) was studied on top of LoRaWAN. FUOTA is a critical requirement for any long-term deployments in order to maintain optimal, safe, and secure operations of the network. However, LoRaWAN limitations such as duty cycle, downlink capability, etc. challenge supporting FUOTA. Consequently, A FUOTA process is proposed, exploiting the new specifications to support multicast, fragmentation, and clock synchronization. The proposed FUOTA process is evaluated to quantify the impact of different parameters on the overall performance. This evaluation helped to determine the best FUOTA parameters.

List of Contributions

The contributions of this thesis have been published in several leading IEEE and ACM conference and journal papers. In all these papers, I am the first author and my role included conceptualization, methodology, validation, writing the original drafts, writing reviews if necessary, project administration, software, resources and investigation. The other authors contributed to one or more of these roles. In addition to that, all the software tools and simulations developed for this thesis have been made publicly available to the research community as open-source.

Thesis Publication

Journal Publications

- **Abdelfadeel, Khaled Q**, Dimitrios Zorbas, Victor Cionca, and Dirk Pesch. Free - Fine-grained Scheduling for Reliable and Energy Efficient Data Collection in LoRaWAN. *Internet of Things Journal*, vol. 7, no. 1, pp. 669-683. IEEE, Jan. 2020.

Conference Publications

- **Abdelfadeel, Khaled Q**, Tom Farrell, David MacDonald, and Dirk Pesch. How to make Firmware Updates in LoRaWAN Possible. *To be Presented In 2020 IEEE 21st International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*, pages –. IEEE, 2020.
- **Abdelfadeel, Khaled Q**, Victor Cionca, and Dirk Pesch. Dynamic Context for Static Context Header Compression in LPWANs. *In 2018 14th International Conference on Distributed Computing in Sensor Systems(DCOSS)*, pages 35–42. IEEE, 2018.
- **Abdelfadeel, Khaled Q**, Victor Cionca, and Dirk Pesch. Fair Adaptive Data Rate Allocation and Power Control in LoRaWAN. *In 2018 IEEE 19th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*, pages 14–15. IEEE, 2018.
- **Abdelfadeel, Khaled Q**, Victor Cionca, and Dirk Pesch. LSCHC: Layered Static Context Header Compression for LPWANs. *In Proceedings of the 12th Workshop on Challenged Networks (CHANTs)*, pages 13–18. ACM, 2017.

Poster Publications

- **Abdelfadeel, Khaled Q**, Victor Cionca, and Dirk Pesch. Poster: A Fair Adaptive Data Rate Algorithm for LoRaWAN. *In Proceedings of the 2018 International Conference on Embedded Wireless Systems and Networks (EWSN)*, pages 169–170. Junction Publishing, 2018.

Other Publications

Journal Publications

- Dimitrios Zorbas, Christelle Caillouet, **Abdelfadeel, Khaled Q**, and Dirk Pesch. Optimal Data Collection Time in LoRa Networks: a Time-Slotted Approach. *Under Review in the Internet of Things Journal*.
- Dimitrios Zorbas, **Abdelfadeel, Khaled Q**, Panayiotis Kotzanikolaou, and Dirk Pesch. Ts-lora: Time-slotted LoRaWAN for the Industrial Internet of Things. *Computer Communications*, vol. 153, pp. 1-10. Elsevier, Mar. 2020.

Conference Publications

- **Abdelfadeel, Khaled Q**, Yasantha Samarawickrama, and Victor Cionca. How to Conduct LoRaWAN Site Surveys. *In 2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 133–138. IEEE, 2019.
- Dimitrios Zorbas, **Abdelfadeel, Khaled Q**, Victor Cionca, Dirk Pesch, and Brendan O'Flynn. Offline Scheduling Algorithms for Time-slotted LoRa-based Bulk Data Transmission. *In the IEEE 5th World Forum on Internet of Things (WF-IoT)*, pages 1–6. IEEE, 2019.

Open Source Software

- Implementing LSCHC (Header Compression Engine) over Contiki OS [1]
- Implementing FADR (A comprehensive LoRa Simulator) using SimPy [2]

[1] `https://gitlab.com/kqorany/SCHC`
[2] `https://github.com/kqorany/FADR`

- Implementing LoRaFREE (A Simulator for Time-slotted MAC on top of LoRaWAN) using SimPy [3]
- Implementing FUOTASim (A FUOTA Simulator for LoRaWAN devices) using SimPy [4]

[3] `https://github.com/kqorany/FREE`
[4] `https://github.com/kqorany/FUOTASim`

Chapter 1

Introduction and Context

The Internet of Things (IoT) term was coined in 1999 by Kevin Ashton in order to link the new idea of Radio Frequency IDentification (RFID) to the Internet [A+09]. He got the required attention for RFID but he did not expect that he had summed up one of the most emerging topics of technical, social, and economic significance. The IoT has no single or universal definition, however, it generally refers to scenarios when the Internet connectivity is extended into the physical world (*i.e.* devices [1]/things) in order for these devices to transfer data over networks without any human intervention [AIM10]. To realize this vision, those devices have to be equipped with sensing or/and actuation capabilities and communication functions commonly wireless. Since this early definition of the IoT, it has been enabling a lot of novel applications from home automation to military applications, health monitoring, smart cities, etc., which has been revolutionizing our way of interaction, living, playing, and working [AIM17].

Subsequently, other terms have been also coined to emphasize the ongoing transformation of IoT, including Internet of Everything and Industrial Internet. Due to the enormous potential impact of IoT, it has been receiving significant attention from academia and industry. In 2017, Cisco estimated that 14 billion devices will be connected to the Internet by 2021 [Cis17]. In the same year, Ericsson went even further in this vision, predicting 18 billion connected devices by 2022 [H+17]. Regardless of which of these two predictions is more accurate, the ground truth is that the volume and the revenue of the IoT industry are in continuous growth [Eri16, L+16]. This is supported by remarkable developments in cheap sensor and actuation technologies and in low-power

[1] Device and node are used interchangeably in this thesis

Micro-controllers and System on Chips (SoCs) that have been enabling complex applications. In addition to the emergence of novel wireless communication technologies that have been helping to extend the range of IoT use cases.

1.1 Modern Wireless Connectivity Landscape of the IoT

A big portion of the devices in IoT refers to small embedded systems that are constrained in terms of memory, computation capability, and energy source. For this, industrial alliances and Standards Developing Organizations (SDOs) have been putting a lot of effort into developing low-power wireless connectivity solutions to met the limitations of those devices. Nowadays, the IoT connectivity landscape is constantly and rapidly evolving, with new technologies being regularly proposed, and with existing ones moving into new application domains. The connectivity landscape can be divided into two categories: short-range and long-range, where the second has recently emerged. Each category reflects a diversity of application domains, which will be highlighted throughout the next sections [PAV+12, AGP+15].

1.1.1 Low Power Short Range Technologies

The propagation range of the technologies in this category varies from a few meters to one kilometre at maximum. Typically, these technologies rely on a multi-hop mesh network topology to extend the communication range beyond the propagation range. In a multi-hop topology, the source can use other devices in the network as relays to reach the destination. In the past two decades, short-range was the dominant technologies for many IoT applications with a lot of effort from the research community to produce low-power mesh network solutions. In the following sections, the most well-known and promising short-range technologies are highlighted.

1.1.1.1 IEEE802.15.4-based Technologies

In 2003, the Institute of Electrical and Electronics Engineering (IEEE) published the IEEE802.15.4 standard, which defined short-range (up to 150 meters) low-power PHYsical (PHY) and Medium Access Control (MAC) protocols. The standard was revised again in 2006 and since then it has been adopted

in many IoT technologies [80206]. ZigBee[2] was one of the first technologies to adopt IEEE802.15.4-2006 specifications to target home automation application [GYYL09]. However, due to the single channel nature of IEEE802.15.4-2006 MAC layer, ZigBee transmissions were unreliable in the multi-hop scenarios, especially with the existence of external interference [LSH08]. An alternative MAC approach that uses synchronized channel-hopping, called Time Synchronized Mesh Protocol (TSMP) was introduced later [PD08]. TSMP was designed primarily for industrial applications that have a stringent reliability requirement. TSMP showed better performance than IEEE802.15.4 in terms of reliability and robustness against interference. Later, TSMP became the foundation of WirelessHART and ISA100.11a [LLW+17]. WirelessHART [3] and ISA100.11a [4] are popular technologies in the domain of process measurement and control applications [PC11]. However, being closed standards, wirelessHART and ISA100.11a products from different vendors caused a lot of interconnection problems. In 2011, the time-synchronized channel hopping technique was adopted in IEEE802.15.4, resulting in the IEEE802.15.4-2011 standard. In the following year, an amendment of the IEEE802.15.4-2011 MAC was published as IEEE802.15.4e [WWDS15], defining three new MAC modes. Among them, the Time Slotted Channel Hopping (TSCH) mode is the most promising, facilitating energy efficient multi-hop communications while reducing interference [WPG15]. Recent deployments of TSCH demonstrated 99% end-to-end reliability in harsh real world conditions [VGZK16].

1.1.1.2 Bluetooth Low Energy (BLE)

In 2010, the Bluetooth Special Interest Group (SIG) proposed BLE as a low-power version of Bluetooth in order to make the protocol suitable for IoT [Blu10]. BLE is designed for short-range communication (up to 50 meters) mainly to target scenarios where user-to-machine interactions are needed, i.e., smart building, healthcare, etc [GOP12]. BLE has the potential to be a *de facto* standard for low-power short-range IoT, considering the fact that more than one billion smartphones are shipped every year, all equipped with BLE interfaces. Since its advent, Bluetooth, and then BLE, supported only a single-hop topology, with one master device communicates with several slave devices [YXL12]. However, in 2017, the Bluetooth SIG announced the mesh

[2] https://zigbeealliance.org/
[3] https://fieldcommgroup.org/
[4] https://www.isa.org/isa100/

networking capability to extend the communication range and simplify the deployments for IoT applications [Blu17]. In addition to that, the mesh BLE enables machine-to-machine communication that would help the technology to move to new use cases such as industry automation [BRSH18].

1.1.1.3 Low Power WiFi (LP-WiFi)

The IEEE802.11 standard, better known as WiFi, was released in 1997 and since then the standard has been evolving, resulted in different WiFi versions a/b/g/n/h/i/ac/ad/af/ai/aj/aq/az/ax/ay [NCF+14]. The main objective of this evolution was to increase the throughput of the wireless transceivers specified by these standards [GdSCK17]. WiFi is a popular standard for indoor short-range (up to 50 meters) high data rate communications with almost 580 million units shipped annually. However, WiFi has not been applied on a large scale to IoT applications, due to its high energy consumption and its poor scalability, i.e., one access point can only support a limited number of stations. However, in 2017, IEEE802.11ah, also known as Low Power WiFi (LP-WiFi), was published to expand the application domains of WiFi networks to the IoT world [KLKG15]. IEEE802.11ah has been designed to support large numbers of power-constrained devices per single access point, to extend the propagation range, and to support small and infrequent data transmissions [ABB+14]. LP-WiFi is based on the IEEE802.11ac standard, but down-clocked by a factor of 10, which increases the propagation range to one kilometre and prolongs the battery life to a couple of years. It also introduces a novel network architecture approach based on grouping, which allows supporting a large number of devices per single access point. So far, deployments of LP-WiFi are still limited as the standard is quite recent.

1.1.2 Low Power Long Range Technologies

The propagation range of the short-range technologies resulted in them being deployed in multi-hop mesh network topology in order to extend their application's data transmission coverage. Typically, the battery-powered devices that use these technologies are connected in a mesh network topology to reach a repeater mode. The repeaters are also networked in a multi-hop fashion in order to reach a gateway. Gateways are typically connected to the Internet via a wired, e.g. Ethernet connection, or a cellular modem. This complex topology usually creates many deployment challenges [BHG11]. The first

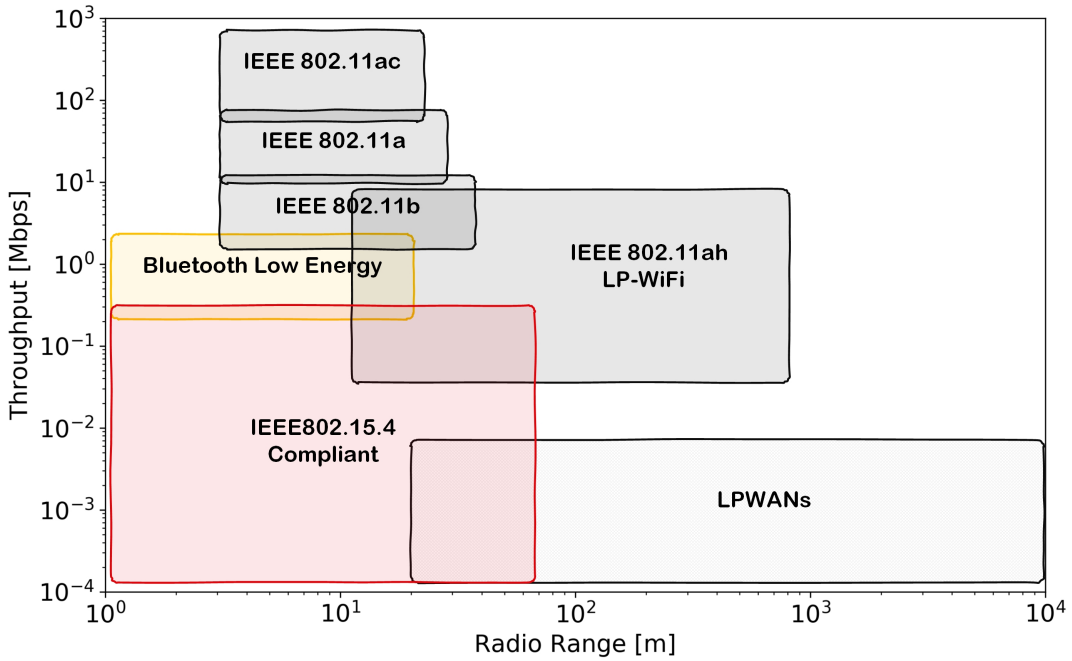


Figure 1.1: LPWANs versus License-free Short-range Technologies

challenge is identifying suitable locations to place the repeaters and the gateways [ACC+08, ABIK06]. Given the many constraints to the problem such as the deployment area constraint, the network design becomes very complex in dense deployments. The second challenge is ensuring reliable and robust connectivity [TXZ05]. The mesh networks usually have a high degree of freedom in routing, which challenges any sort of Quality of Service (QoS) requirements [WBII06]. Also, due to the often highly varying wireless channel dynamics, parts of the network often are in an outage state [YYW+05].

Therefore, we have recently witnessed a shift towards a new communication paradigm that can offer low-power communications and long propagation range at the same time. Using this paradigm, network planning is much simpler as devices can directly connect to the gateways in a simple star topology without the need for repeaters. This new class of communication technologies have emerged lately, termed Low Power Wide Area Networks (LPWANs) [AGP+15, RKS17].

1.2 Unlicensed Low Power Wide Area Networks (LPWANs)

The term LPWANs was first used in 2014 by Machina Research who used the term to describe the new market opportunity for interconnecting a large number of low-cost devices using the emergent technologies that offer long propagation rages [Res14]. In their report, Machina Research also estimated that up to 60% of low-cost devices will be connected using LPWANs by 2022, which would create 3 billion LPWAN connections by 2023. Since then, much hype has surrounded the LPWANs and many researchers have expected that LPWANs would enable the anticipated IoT scale not only in terms of the number of devices but also in the variety of services. Sigfox [5], LoRa [6], Weightless [7], Ingenu [8], DASH7 [9], MIOTY [10], and NB-Fi [11] are some of the many technologies that recently hit the market and can be considered as LPWANs.

LPWANs are unique because they can offer different trade offs than the traditional short-range wireless technologies. Figure 1.1 highlights these differences in terms of throughput and propagation range. LPWANs are not considered to be a replacement of the short-range technologies but a complement to them. The long-range and the low-power attributes of LPWANs usually come at the expense of low data rates (typically in orders of tens of kilobits per seconds) and high latency (typically in orders of seconds or minutes). Consequently, these technologies are not suitable for high throughput applications that require ultra-low latency such as wireless industrial control. Nevertheless, a plethora of applications such as in smart cities, smart metering, home automation, wearable electronics, logistics, environmental monitoring, etc. that only exchange infrequent small amounts of data find LPWANs the perfect option. Therefore, the appeal of LPWA technologies, although limited by their low data rate, is still broad [RKS17].

[5] https://www.sigfox.com
[6] https://www.semtech.com/lora
[7] http://www.weightless.org/
[8] https://www.ingenu.com/
[9] https://dash7-alliance.org/
[10] https://behrtech.com/mioty/
[11] https://waviot.com/technology/what-is-nb-fi

1.2.1 Design Goals of LPWANs

LPWAN technologies share the same set of design goals, where each technology has tried to achieve these goals by taking different design decisions. These design goals can be summarized as follows [RKS17, BKKI20, MBCM19]:

- **Long Range**: LPWANs are designed to offer long-range wireless links up to tens of kilometres in rural environments and about 1-2 kilometres in urban environments such as cities. This gives great freedom to the devices to spread over large geographical areas which is a desirable feature by many IoT applications. Sub-1GHz band and special modulation techniques are employed to achieve this design goal. Most of the LPWAN technologies operate in the unlicensed spectrum of the sub-1GHz band. Transmissions in Sub-1GHz experience less attenuation and multipath fading compared to 2.4GHz, which is used in the traditional short-range WSNs. In addition to that, sub-1GHz is less crowded than 2.4GHz, which is used by many wireless technologies such as WiFi, ZigBee, Bluetooth, etc. This results in more robust wireless links that can enable long-range communications. Novel modulation techniques such as Chirp Spread Spectrum (CSS) in LoRa and Ultra Narrow Band (UNB) in Sigfox are the second reason behind the long-range communications of LPWANs. These modulation techniques can achieve link budgets of 150 ± 10 dB compared to 120 ± 10 dB in the case of the short-range technologies. Consequently, the receivers of LPWAN technologies can decode severely attenuated signals correctly, which also help to achieve long-range communications.

- **Energy Efficiency**: LPWANs aim at low-power operations in order for the connected devices to operate for up to 10 years on a coin battery. Long battery lifetime is also a very desirable feature for many IoT applications as it removes the need to replace the devices' battery. In particular, for devices that are expected to be deployed in inaccessible places or remote areas such as volcanoes [WALJ+06], glaciers [HTB+08], etc. Various techniques are used to ensure the low-power operations. First, All LPWAN technologies operate in simple network topology, i.e., star-topology, where devices connect directly to the gateways without the need for repeaters. This brings huge energy-saving advantages as devices do not have to relay any transmissions from other devices. Consequently, they spend most of their time in a sleep mode and, thus achieving low-energy consumption unlike in a mesh network topology. Secondly, LPWANs usually use simple and lightweight MAC protocols such as ALOHA in which devices transmit whenever they want without performing any sort of carrier

sense. Simple MAC protocols bring very low-power operations. Finally, LPWAN technologies tend to simplify the design of the protocols on the devices by offloading the complexity to the back-end, i.e., network server. Typically, devices in LPWANs do not have to register with a particular gateway, which requires additional signalling overhead. Instead, all gateways in the range of a device's transmission can receive the transmission and then the network server has to filter the duplicate transmissions. Also, the back-end is usually responsible for managing the resources of the devices on their behalf. This removes the burden of running heavy protocols at the devices in order to maximize their link capacities.

- **Low Cost**: Commercial feasibility and, consequently cost is one of the most important factors for the wide-spread adoption of any technology. Therefore, LPWANs are designed with low cost as one of their main design goals. This has been realized by deploying all LPWANs in unlicensed spectrum bands, where no fees or expensive licenses are required to use them. Also, the fact that LPWANs rely on simple star typologies contributes positively to achieving low cost. Specifically, minimum infrastructure is required, i.e., no need for repeaters, which reduces the deployment cost for network operators. In terms of the cost of devices, LPWANs use low-complexity modulation techniques that can be processed using low-cost transceivers. As a result, a typical transceiver of LPWAN technologies costs less than 10 Euros.

- **Scalability**: LPWAN technologies are designed and scalability is in mind in order to operate well even when a network grows to a massive number of devices. Scalability can be interpreted as how many devices a single gateway can handle. For most LPWAN technologies, this number ranges from a couple of hundred to a few thousand, depending on the deployment conditions. In fact, scalability depends directly on the deployed applications and the traffic patterns of these applications, i.e., how frequently a device transmits and receives data. With moderate traffic patterns, LPWANs can support a massive number of devices, which has the potential to truly enable the IoT vision. In order to boost the scalability, most LPWANs depend on different techniques of access diversity, for example, channels and data rates to accommodate as many devices as possible. For example, In LoRa, multiple quasi-orthogonal spreading factors are supported. Specifically, two transmissions can overlap in time and frequency but if they use different spreading factors, the receiver can decode both transmissions correctly with a high probability.

With these ambitious design goals, LPWANs have the ability to address diverse requirements of IoT applications such as smart metering, monitoring applications, e.g., fire and intruder alarms applications, and smart city applications [HDPMH18]. Therefore, LPWAN technologies are considered as a potentially key driver for the global adoption of IoT.

1.3 Cellular LPWAN Technologies

As a reaction to the growing popularity of the unlicensed LPWAN technologies, 3rd Generation Partnership Project (3GPP [12]), the standards organization that develops protocols and systems for mobile telecommunications, has put a lot of effort into addressing these market requirements as well. Cellular technologies such as Long Term Evolution (LTE) have been designed without considering the link budget requirements for IoT devices or their traffic patterns, resulting in power-hungry and poorly scalable IoT applications [Mis07, PDG+16]. In order to support the requirements of IoT applications, several improvements were proposed to the existing cellular technologies to strip complexity, reduce cost, and improve range and battery lifetime. The LTE-M standard, where M stands for machines, was proposed as an optimized version of the LTE standard dedicated for long-range communications to meet IoT application requirements [LKM+16]. Reducing data rate, adopting Power Saving Mode (PSM), and Extended Discontinuous Reception (eDRx) are some of the techniques used to reduce the cost of LTE-M devices, to extend communication coverage, and to prolong the battery life. These improvements extend the battery life to up to a few years for downlink delay-tolerant traffic, reduces the complexity of the devices by 50%, and increases the coverage by 15-20 dB compared to the LTE technology [DZGPRPM16]. The LTE-M standard is compliant with the LTE standard, where both technologies can coexist side by side on the same frequency bands and can be deployed using the same infrastructure.

In addition to that, 3GPP proposed a new technology that is primarily optimized for IoT applications, called Narrow Band IoT (NB-IoT) [WLA+17]. Table 1.1 shows some of the difference between NB-IoT and LTE-M. Narrow Band IoT is a new radio access system built on some existing LTE functionalities with essential simplifications and optimizations. NB-IoT has offered more freedom in terms of the design decisions, resulting in further cutting in device cost and

[12] https://www.3gpp.org/

Table 1.1: The Differences between NB-IoT and LTE-M

	NB-IoT	LTE-M
Packet Data Rate	< 100 Kbps	384 Kbps - 1 Mbps
Latency	1.5 - 10 s	50 - 100 ms
Power Consumption	Best at very low data rate	best at medium data rate
Mobility	Limited	Yes
Voice (VoLTE)	No	Yes
Antennas	1	1

energy consumption compared to LTE-M. Besides the optimization in LTE-M, NB-IoT has specified two new optimizations techniques for small data transmission [RMZ+16]. The first one is the ability to transmit small amounts of data in the control plane via Signaling Radio Bearer (SRB). The second one is the ability to suspend and resume a Radio Resource Control (RRC) connection and, thus eliminating the need to establish a new connection at each reporting instance. Consequently, NB-IoT can achieve 164 dB link budget along with a data rate of 0.40 kbps in the downlink and 0.27 kbps in the uplink. In terms of scalability, a few thousands devices within a cell-site sector can be supported without sacrificing the end-to-end delay, which is expected to be 10 seconds or less for 99% of the devices [RMZ+16]. NB-IoT requires only 180KHz bandwidth and, thus cellular operators can re-farm their Global System for Mobile Communications (GSM) or LTE spectrum to deploy NB-IoT.

Figure 1.2 [13] represents a qualitative comparison of the cellular LPWAN technologies with two promising technologies from the unlicensed world, LoRa and Sigfox. The compassion shows different metrics, including Firmware Updates Over The Air (FUOTA) support, IP interoperability, latency, cost efficiency, scalability, propagation range, popularity (i.e., number of deployments), device lifetime and data rate.

1.3.1 Advantages of Cellular LPWAN Technologies

The designing goals behind LTE-M and NB-IoT technologies are similar to the ones discussed previously for the unlicensed LPWAN technologies. However, the licensed spectrum and the ecosystem of 3GPP gave the cellular LPWAN technologies extra privileges that cannot be found in the unlicensed LPWAN technologies.

[13] Radar charts are used to examine the relative values for a single data point (e.g., Data Rate is the highest for LTE-M and the lowest for Sigfox)

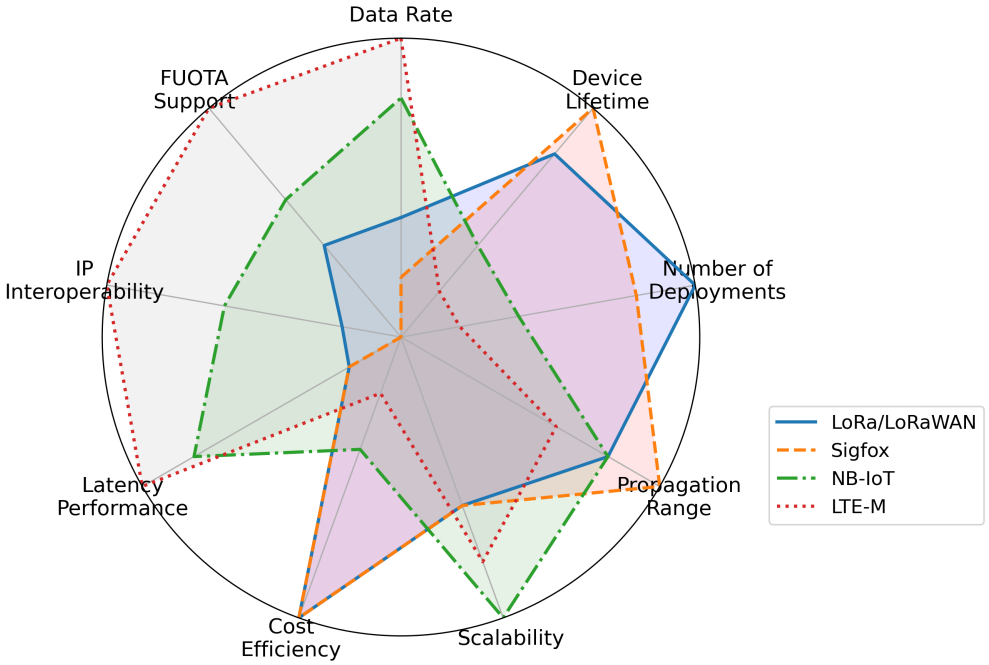


Figure 1.2: Cellular LPWANs versus LoRaWAN and Sigfox

• **Quality of Service** (QoS): The ability to guarantee a minimum level of QoS could be a killer feature for any LPWAN technology. Achieving this is relatively easy in the licensed spectrum because of the controlled interference unlike the case in the unlicensed spectrum where many technologies share the same spectrum. In addition to that, this exclusivity of spectrum use gives more freedom to use higher transmit power levels that can help to achieve the required QoS. In contrast, technologies operating in the unlicensed spectrum must adhere to the duty cycle and the transmitted power regulations of the spectrum.

• **Internet Interoperability**: Most unlicensed LPWAN technologies have adopted a simple network stack that does not support Internet Protocol (IP) functionalities. In this case, the devices of the unlicensed LPWANs cannot understand the Internet language, which may limit the application domains of these technologies. To address this issue, IoT middleware or back-end solutions are used to achieve Internet interoperability. In the licensed LPWAN technologies, except for NB-IoT, the network stack is more mature to support IP and, thus it is easier to achieve a seamless integration with regular Internet services.

• **Firmware Updates Over The Air** (FUOTA): All unlicensed LPWAN technologies do not support FUOTA, which would pose great security risks on the deployments of these technologies. In contrast, cellular operators provided

proven FUOTA mechanisms for their traditional cellular technologies that are easy to adopt for cellular LPWAN technologies. FUOTA was not available for cellular LPWAN technologies at the time of writing the thesis however in principle, this is easier to be performed in cellular networks than unlicensed LPWAN technologies.

1.4 Future of LPWANs Market

There are different opinions about the future of the LPWANs market and whether the market will tend towards the unlicensed or the licensed technologies [PAV+12, RKS17]. The ground truth in this matter is that the unlicensed LPWAN technologies, for example, LoRa/LoRaWAN and Sigfox, have hit the market a couple of years ahead of the licensed technologies. This gave these technologies a time advantage to deploy their technologies without real competition from the cellular operators. For example, Sigfox with its national deployment business model is already available in over 70 countries and regions and aims to cover 100% of the globe in the next few years [14]. While with initiatives such as The Things Network [15] along with many private deployments, LoRa/LoRaWAN devices are deployed in 140 countries on every populated continent [16].

Given the size of the LPWAN market, to the belief of the author, there are market shares for many LPWAN technologies side by side from both the unlicensed and the licensed worlds. The unlicensed LPWAN market share, of course, will be affected by the licensed technologies entering the market, especially, when QoS is a crucial requirement for certain applications. However, as long as the unlicensed LPWAN technologies target the right applications that suit their features, they might not be displaced by the licensed technologies.

1.5 Scope, Motivation, and Goal of the Thesis

Since the early days of LPWANs, a major hype has surrounded them, making it sometimes difficult to clearly understand and assess their capabilities. This has been inflated with biased reports and inaccurate data to promote even inapplicable solutions. When this thesis was started in late 2016, the scientific

[14] https://www.sigfox.com/en/coverage
[15] https://www.thethingsnetwork.org/
[16] https://lora-alliance.org/lorawan-coverage

research around the LPWANs was still in its infancy because of the recency of the topic. The published works mainly focused on evaluating the physical layer of these LPWAN technologies by performing experiments with a small number of devices. The objective of these experiments was to report different attributes of the physical links such as data rate, range, power consumption, etc in order to investigate the differences among these new communication links [WKL+16, MPH16, GVN16, PMHI16, OGS17, JKM+17]. Only a few works have focused on the upper layers, e.g., MAC layer, of these technologies to understand the network performance in terms of scalability, reliability, etc. [BKL17, GR17, MCV17]. Most of the published works looked at LoRa/LoRaWAN technology because of its suitability for private deployments, e.g., on a university campus. In such cases, researchers have full control over all the network entities. Unlike the other LPWAN technologies, e.g., Sigfox, that run through a single national coverage.

The goal of this thesis was set to study the network stack of the LPWAN technologies, and specifically LoRaWAN, in different scenarios in order to understand their advantages and disadvantages in supporting IoT applications. Consequently, proposing improvements in order to enhance their performance and to extend their application domains, taking into account all network limitations. Specifically, resource management and IP interoperability topics were the main focus of the research because there was almost no work around these two topics in the LPWAN domain. Within these topics, this thesis tries to answer the following research questions:

- **How to ensure interoperability among LPWANs themselves and between them and the Internet?** At the device level, there is no common language among LPWANs and the regular Internet. This is because all LPWAN technologies rely on a simple network stack, where applications typically are built on top of the MAC layer directly. This harms the vision of the IoT, which calls for seamless integration with the Internet. In other words, LPWANs' devices have to support IP functionalities. This approach is also inline with the work that was done for ZigBee technology, i.e., IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN), to ensure Interoperability with the Internet services. However, the situation here is different because LPWANs are a more constrained class of networks in terms of data rates, maximum frame lengths, etc. than the ZigBee-like networks. Consequently, the research plan here was set to investigate how IP functionalities can be supported on top of LPWANs, and eventually supporting the full native Internet Engineering Task Force (IETF)

IoT stack, i.e., IPv6/UDP/Constrained Application Protocol (CoAP) for LPWANs devices.

• **Is LPWAN, and especially LoRaWAN, a fair system? If not, how can fairness be improved?** The research plan here was set to study the performance of the network in terms of energy consumption, Data Extraction Rate (DER), scalability, and etc. In particular, assessing the fairness of the system. In addition to that, understanding what factors impact these metrics. Consequently, proposing algorithms/protocols to improve the performance. One of the major advantages of LPWANs is the ability to connect a very large number of devices without degrading the DER. But how many devices roughly without negatively affecting the fairness of the system? As this number is expected to be large, studying the network performance through real-world experiments would not be feasible. Alternatively, computer simulation is a good tool to overcome the lack of real resources. Nevertheless, for the study to be fairly accurate, good modelling of the PHY and the MAC layers of the existing LPWANs are required. Therefore, another research plan was set to build an accurate simulation tool in order to assess the network performance.

• **What is the performance of LPWANs, and especially LoRaWAN, for monitoring applications in remote areas? How to improve network performance?** In order to answer these questions, the research plan was set to understand the overall performance of LPWANs in terms of reliability, energy consumption, and scalability under the requirements from monitoring applications in remote areas such as environmental monitoring and agriculture monitoring. The monitoring applications in remote areas are expected to support a lot of devices. In addition to that, the applications may require the devices to *periodically* or *a-periodically* update their status to the back-end (e.g., the gateway and the network server). These type of applications are argued to be the best fit for the LPWAN technologies, and especially LoRaWAN, considering their wide coverage and low-power consumption that do not require human intervention for battery replacement for example. However, remote areas (e.g., lack of cellular coverage or infrastructure) challenge the performance of the network in terms of reliability, scalability and energy consumption. From this study, new solutions are proposed to address the requirements of monitoring applications in remote areas.

• **How to make Firmware Updates over LPWANs, and especially LoRaWAN, possible?** Supporting FUOTA on top of LPWANs is a crucial feature

for their long-term deployments. With FUOTA's support, security updates, new functionalities, and optimization patches can be deployed with little human intervention to the LPWANs' devices over their lifetime. However, FUOTA is a challenging task in LPWANs because these networks are designed mainly for uplink transmissions, not heavy downlink transmissions that are required for FUOTA. Therefore, the research plan was set to investigate possible solutions to enable FUOTA on top of LPWANs without sacrificing the low-power operations concept of LPWANs.

1.6 Contributions

The contributions of this thesis have resulted in novel algorithms/protocols and improvements, which target different network layers of LPWANs stack, from the MAC layer to the application layer. These contributions have also been published in leading IEEE and ACM conferences such as IEEE DCOSS, IEEE WoWMoM, IEEE World Forum on the Internet of Things, ACM EWSN and ACM CHANTs as well as the leading IEEE Journal on the Internet of Things and Elsevier Computer Communication. These contributions can be summarized as follows:

- **Improving the performance of the Static Context Header Compression (SCHC) protocol**. SCHC is a new stateless header compression protocol that has been proposed by the LPWAN IETF working group for enabling IPv6 traffic over LPWANs. First, SCHC was implemented over the Contiki operating system in order to evaluate its performance. Second, some simulations are conducted over Cooja simulator in order to compare the compression efficiency of SCHC against the IPHC protocol (the protocol used in 6LoWPAN). Although SCHC shows higher compression efficiency, its memory usage is higher compared to IPHC.

Consequently, a new technique of storing the SCHC rules was proposed, which has the potential to reduce the overall memory footprint when implementing a large number of SCHC rules in constrained devices. Without scarifying the compression efficiency of SCHC. Because SCHC works under the assumption that LPWANs are pre-programmed with known data flows, it uses a static context for compressing the data flows. However, this assumption is not always valid in the IoT context, where devices should be accessible from any IPv6 address (i.e., unknown data flows are expected). Therefore,

a new compression technique was proposed to allow SCHC to effectively compress/decompress some header fields of unknown data flows in order to improve the overall compression efficiency.

These proposals to SCHC enhance its performance and applicability in order to extend the Internet architecture to the LPWANs. In other words, enabling end-to-end IP traffic over LPWANs. This would bring the power of openness, interconnection, cooperation, and standards to LPWANs devices and applications.

• **Improving the fairness of LoRaWAN DER**. In order to understand the network performance of LPWANs, the PHY and the MAC layers of LoRaWAN, as one of he most widely-deployed examples of LPWANs, were studied. This study involved modelling these two layers in a simulator, build on top of the Python SimPy library, along with implementing the adaptive data rate mechanism of LoRaWAN that is responsible for managing the resources in terms of data rate, channels, and transmission power levels. This study revealed that LoRaWAN exhibits several characteristics that can lead to unfair distribution of the DER among devices, leading to poor scalability. First, LoRa modulator causes capture effect, where strong signals suppress weaker signals at the gateway, resulting in unfair data extraction rates. Secondly, LoRaWAN offers various data rates that have unequal transmission times for the same packet length, leading to unfair collision probabilities.

Given these conditions, devices experiencing higher attenuation or using low data rates are less likely to see their packets received correctly. Therefore, a novel fair adaptive data rate protocol is proposed to address these factors of unfairness. Since LoRa/LoRaWAN supports various data rates, the fairest ratios of deploying each data rate within a deployment are derived for a fair collision probability. In order to deal with the capture effect, a transmission power control algorithm is proposed to balance the received signal power levels from all devices regardless of their distances from the gateway for fair data extraction. Simulation results showed that the proposed approaches achieved higher fairness in data extraction rate than the state-of-art in almost all network configurations.

• **Fine-grained Scheduling for Reliable and Energy Efficient Data Collections in LoRaWAN**. Studying the performance of LoRaWAN for bulk data collection applications revealed that LoRaWAN suffers from poor scalability when trying to support a large number of devices. The main reason behind that is

the high collision probability of the Aloha-based MAC layer of LoRaWAN. The scalability even worsens further when using acknowledged transmissions due to the duty cycle restriction at the gateway. Therefore, a novel time-slotted MAC protocol along with a new adaptive data rate mechanism are proposed. The new protocols can be easily implemented on top of LoRaWAN.

The new proposal takes advantage of applications that do not have hard delay requirements on data delivery by supporting synchronized bulk data transmission. This means data have to be buffered for transmission in scheduled time slots instead of transmitting it straight away. The new proposal allocates spreading factors, transmission powers, frequency channels, time slots, and scheduled slots in frames for LoRaWAN devices. As a result, the proposal overcame the scalability problem of LoRaWAN by eliminating collisions and grouping acknowledgements. The numerical results from the performance evaluations showed that the new proposal scales well and achieves almost 100% data delivery and the device lifetime is estimated to over 10 years independent of traffic type and network size. Comparing to poor scalability, low data delivery and device lifetime of fewer than 2 years for acknowledged data traffic in the standard LoRaWAN.

• **Enabling firmware updates over LoRaWAN**. Embedded software management requirements due to concerns about security vulnerabilities or for feature updates in IoT deployments have raised the need for FUOTA. With FUOTA's support, security updates, new functionalities, and optimization patches can be deployed with little human intervention to embedded devices over their lifetime. However, supporting FUOTA over LoRaWAN is not a straightforward task due to LoRaWAN's limitations that do not provide downlink bulk data transfer such as a firmware images.

Therefore, the LoRa Alliance has proposed new specifications to support multicast, fragmentation, and clock synchronization, which are essential features to enable efficient FUOTA in LoRaWAN. These new specifications are reviewed and the FUOTA process is evaluated in order to quantify the impact of the different FUOTA parameters in terms of the firmware update time, the device's energy consumption, and the firmware update efficiency, showing different trade-offs among the parameters. This study determined the best FUOTA parameters for the aforementioned metrics.

1.7 Structure of the Thesis

The remainder of this thesis is organized as follows:

Chapter 2 highlights some of the details about LPWANs, specifically, LoRaWAN that will be referred to frequently throughout this thesis.

Chapter 3 overviews SCHC and details the proposed improvements over the protocol to achieve less memory footprint and higher compression ratio. This chapter also shows the simulation campaigns and the analytical models that are used to measure the performance.

Chapter 4 shows the study of the PHY and the MAC layers of LoRaWAN along with the proposed resource allocation protocol to achieve fair data extraction rate in LoRaWAN. In addition to that, simulation results are given in this chapter to show the advantages of the proposed protocol against the state-of-the-art in terms of fairness and energy consumption.

Chapter 5 details the proposed MAC protocol that together with the proposed new resource allocation protocols have the potential to achieve scalable and reliable data collection applications. This chapter also presents extensive simulation results that show the performance of the proposals compared to the standard LoRaWAN.

Chapter 6 reviews the multicast, the fragmentation, and the clock synchronization specifications by LoRa Alliance that are essential to perform FUOTA. In addition to that, this chapter shows how to make FUOTA possible over LoRaWAN and what the impact of the different FUOTA parameters on the whole process.

Chapter 7 concludes the research work and highlights the most important takeaway messages. This chapter also discusses the future work that can be pursued based on this thesis.

Chapter 2

A Preliminary to LPWANs

Chapter 1 highlighted the goals behind designing the LPWAN technologies a) long-range coverage, b) energy-efficient operations, c) low-cost devices and deployments, and d) scalable access. LPWANs balance battery lifetime, cost benefits , and wide area connectivity at the expense of data rate by operating a narrow bandwidth and with limited duty cycle.

These design goals are found to be suitable for a wide rage of use cases from environmental monitoring to smart cities, resulting in many novel technologies. Sigfox [1], LoRa [2], Weightless [3], Ingenu [4], DASH7 [5], and NB-Fi [6] are some of many technologies that operate in the unlicensed spectrum. In addition to that, LTE-M and NB-IoT are designed mainly to operate in the licensed spectrum band.

2.1 A Typical LPWAN Architecture

Almost all LPWANs have similar network architecture as shown in Figure 2.1 [Far18]. A typical LPWAN consists of four components: *end-devices*, *gateways*, a *network server*, and *application servers*. Each network uses its own terminology, for example, end-devices are called users equipment in the context of NB-IoT. From here on, the terminology of LoRa/LoRaWAN will be used.

End-devices (devices for short) are the majority of components in a network

[1] https://www.sigfox.com
[2] https://www.semtech.com/lora
[3] http://www.weightless.org/
[4] https://www.ingenu.com/
[5] https://dash7-alliance.org/
[6] https://waviot.com/technology/what-is-nb-fi

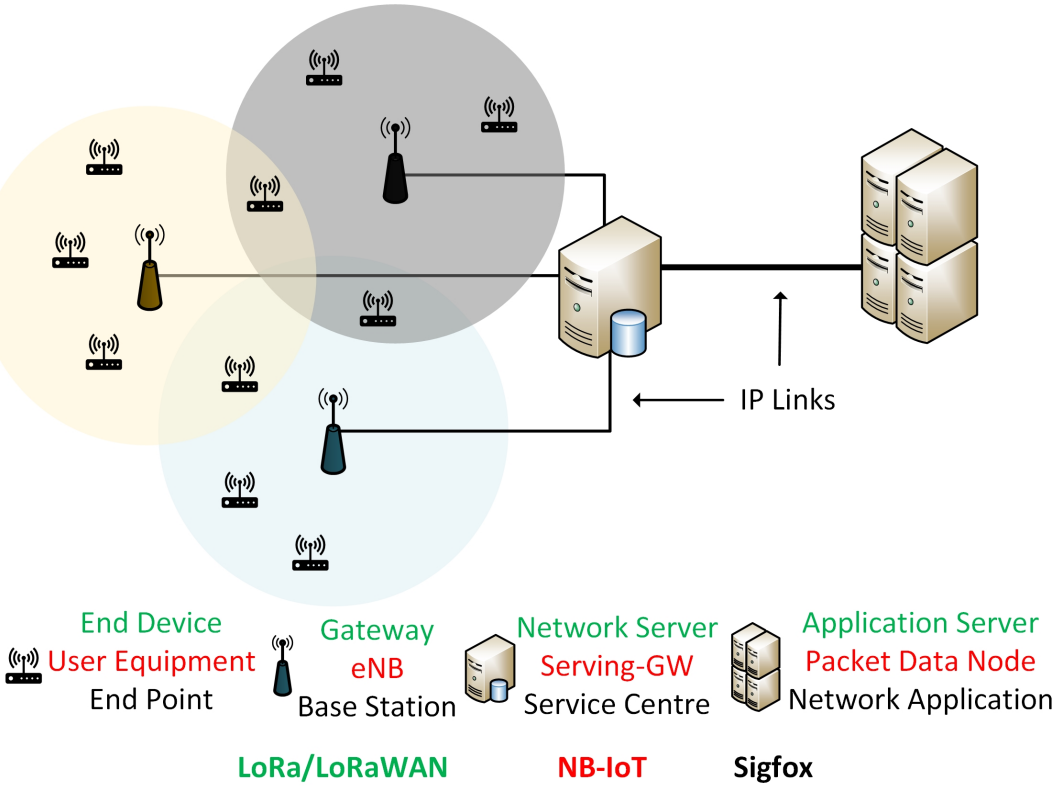


Figure 2.1: Similar Architecture but Different Terminologies

and they are typically equipped with sensors to perform sensing functions, e.g., temperature, occupancy, etc. The aim here is to send the collected sensing data to an application server. In other scenarios, the devices are equipped with actuators for actuation functions, e.g., electric motor, solenoid, etc. In this case, the devices receive commands from an application server to execute a certain task. In some cases, the devices are performing both functionalities, i.e., sensing and actuation. For sending and receiving functionalities the devices are equipped with wireless radio modules of one of the LPWAN technologies. The devices should be deployed in the coverage of one or more gateways at fixed positions or with limited mobility.

The gateways are also equipped with radio modules from the same technology as the devices. But usually the radio modules of the gateways are more powerful in terms of signal processing in order to decode multiple signals simultaneously. All gateways of the same network are connected to the same component, i.e., network server via IP links, typically using cellular networks such as 3G or 4G. Gateways receive/transmit packets from/to devices in their coverage and, thus they represent the end of the constrained links. Typically, gateways run on a

minimal firmware, i.e., packet forwarder, to make them low-cost and easy-to-use. The packet forwarder encapsulates the packet received from devices in IP/TCP packets to be forwarded to the network server. Gateways also perform the opposite at the downlink when the network server sends something to the devices. Gateways are typically installed at fixed locations, but they can also be mobile, e.g., on drones [ZO18].

The network server is the brain of a network as it carries the network intelligence such as resource allocation. The network server terminates the MAC layer of the network and it takes care of interpreting and collecting the uplink transmissions. The functions of a network server can be summarized as follows:

- **Redundancy Handling**: Multiple copies of the same uplink transmission may reach the network server via multiple gateways. In this case, the network server relays only the strongest packet to the corresponding application server.
- **Downlink Routing**: In case of a downlink transmission, the network server picks the best gateway through which to send data to the intended device. Typically, this decision is based on the link quality, calculated using RSSI and SNR of previous uplink transmissions.
- **Network Control**: The network server handles the provisioning and the security functions of the different network entities. In addition to that it controls the resources in terms of data rate, channel(s), transmission power levels, and etc. of each device individually for low-power consumption and/or network optimization.
- **Gateway Supervision**: The network server allows the network operator to monitor the connected gateways, handle faults and alarms.

Devices in a network may support one or more applications, which refers to the code at the application layer of the devices. An application server manages single or multiple applications over single or multiple networks. Application servers run behind the network server via IP links and provide software frameworks that enable creating applications and server environments to run them. In addition to that, the application servers may provide useful statistics about the running applications.

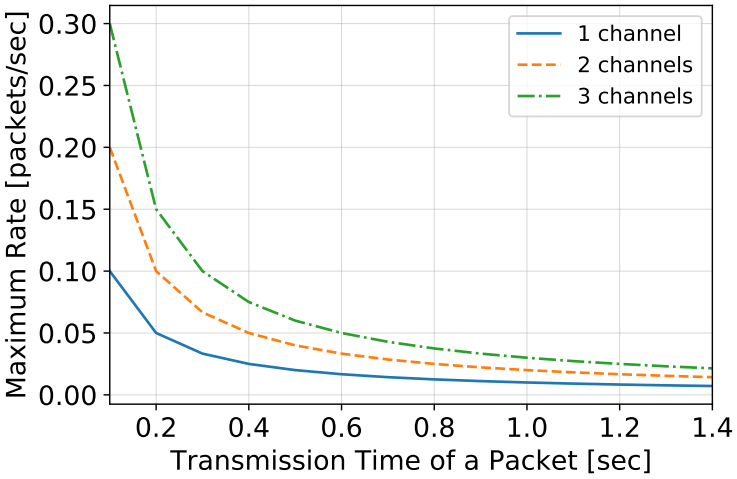


Figure 2.2: Maximum Transmission Rate under 1% Duty Cycle

2.2 The Sub-1GHz band

Most of the unlicensed LPWAN technologies use the sub-1GHz band, which offers excellent signal propagation and robustness against interference at a low power budget. In the sub-1GHz spectrum, LPWAN technologies typically use the 868 MHz band in Europe and the 915 MHz band in USA. The advantages of the sub-1GHz band over the 2.4GHz band in terms of the propagation range and power budget comes at the expense of more restricted spectrum regulations. For instance, the 2.4 GHz band does not impose duty cycle regulations, but only a restriction on the maximum power level applies. However, the sub-1GHz band applies a limited duty cycle regime plus regulation on the maximum allowable transmission power level. For example, in Europe, 10% duty cycle applies at best and more often 1% for wireless nodes that do not apply a listen-before-talk policy. As most of the LPWAN technologies use simple MAC protocols, they have to obey this limited duty cycle regulations.

The duty cycle of a channel determines how often and for long a wireless module device should transmit on this channel. In other words, if a device transmits a packet on a duty cycled channel, this device can't transmit again on the same channel until some time has passed (silent period). This silent period depends on the transmission time of the last packet sent and the duty cycle of the channel. Equation 2.1 defines that silent period, where d denotes the duty cycle and T_p denotes the transmission time.

T_s as in Equation 2.1.

$$T_s = T_p(\frac{1}{d} - 1)[sec] \tag{2.1}$$

Consequently, a 1% duty cycle results in a maximum transmission time of only 36 seconds per hour for each device using that channel. In order to increase this transmission rate, devices transmit on different channels in different sub-bands. In this case, the maximum transmission rate would equal

$$\frac{nd}{T_p}, \tag{2.2}$$

where n is the number of channels. Figure 2.2 shows the maximum transmission rate while observing the 1% duty cycle for using up to three channels. As shown, for long packets (transmission time equals ca. 1.4 secs) a device would only be allowed to transmit 26 packets per hour at maximum. For this reason, a number of LPWANs such as Ingenu still exploits the 2.4 GHz band to avail of more relaxed spectrum regulations on radio duty cycle.

2.3 LoRaWAN Technology

This section provides an overview of LoRaWAN, one of the better known LPWAN technologies. LoRAWAN has been receiving a lot of attention from industry and academia and has become a popular LPWAN technology. LoRaWAN supports an open business model, where the specifications of the technology are publicly available and its hardware are fairly cheap and available. These were the main reasons for researchers to consider this technology for further research. The latest LoRaWAN coverage report highlights that LoRaWAN networks are deployed in 143 countries with more than 10 million devices and 133 LoRaWAN operators.

Many other technologies exist in the market but most of them are proprietary solutions. Additionally, there is insufficient information publicly available on them to allow proper research on them. Chapters 4, 5, and 6 focus on research problems in LoRaWAN and thus this section provides an overview about the technology.

LoRaWAN is a wireless technology that is built upon LoRa modulation to target low-power, long-range, and low-data rate applications [LoR17b]. LoRaWAN defines the MAC rules, the system architecture, and the regional parameters for operation in different regions of the world. While LoRa defines the physical

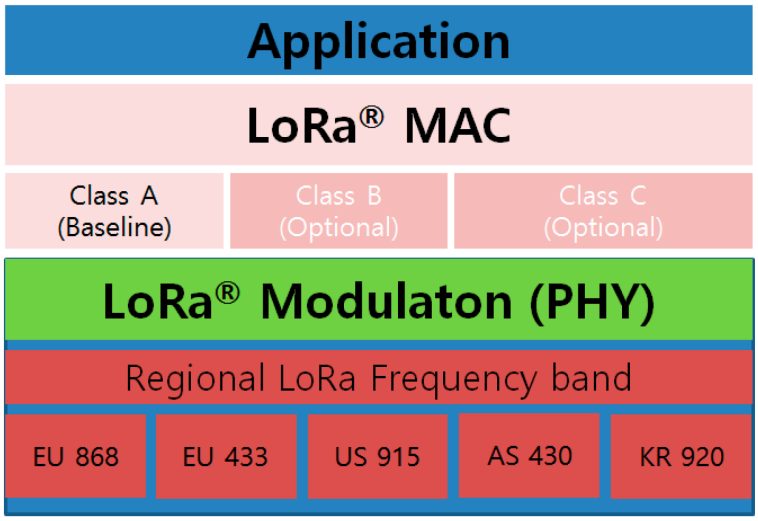


Figure 2.3: LoRaWAN Stack [LoR17b]

layer for the sub-1GHz radio band. LoRaWAN technology was defined by the LoRa Alliance[7] and formalized in specifications, which are publicly available.

LoRaWAN follows the same network architecture (star topology) as described in Section 2.1. In which, devices communicate directly with all gateways in the communication range. Thus, gateways of the same network are connected to the same network server. LoRaWAN stack is simple and consists only of three layers, where applications are sitting right on top of the MAC layer as shown in Figure 2.3.

LoRa [Sem15, Van17] (short for **Lo**ng **Ra**nge) is a proprietary low-cost implementation of Chirp Spread Spectrum (CSS) technology by Semtech [8]. The first use of CSS technology for IoT was developed by Cycléo, a French company that was acquired by Semtech in 2012. CSS is not a new technology as it is widely used in radar systems and was first proposed for communication systems by Winkler in 1962 but had been barely used since [Win62].

2.3.1 CSS Operation in LoRa

CSS uses wide band linear frequency modulated pulses, called *chirps* to encode data symbols. Thus, CSS is a subcategory of Direct-Sequence Spread Spectrum (DSSS). DSSS benefits from controlled frequency diversity, which allows recovering signals even from below the noise floor to meet the needs of IoT. This is

[7] A non-profit association of more than 500 member companies, committed to enabling large scale deployment of LoRaWAN <https://www.lora-alliance.org/>

[8] https://www.semtech.com/

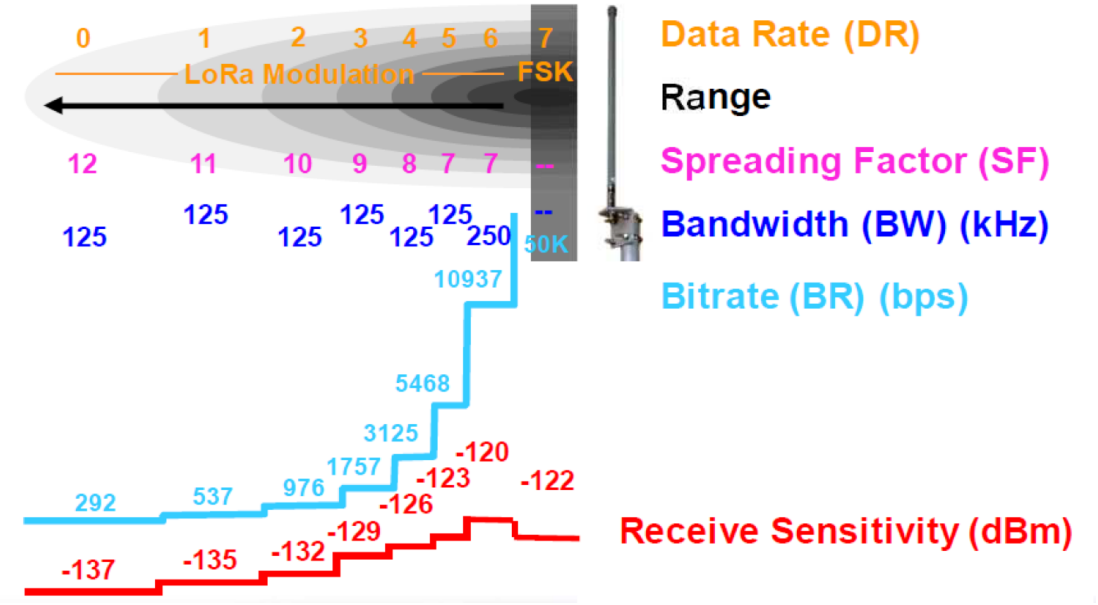


Figure 2.4: LoRa Modulation Parameters [Sem15]

achieved by trading data rate for sensitivity, resulting in long communication range.

A LoRa symbol covers the entire bandwidth, making the modulation robust to channel noise and insensitive to frequency shifts. LoRa modulation is defined by two main parameters: Spreading Factor (SF) $f \in (7, ..., 12)$, which defines the number of encoded bits per a symbol, and Bandwidth (BW) $b \in (125, 250, 500)KHz$, which is the spectrum occupied by a symbol. A LoRa symbol consists of 2^f chips in which chip rate equals bandwidth. Thus, increasing the SF increases the symbol time and, thus increases the transmission time of a packet. LoRa uses forward error correction codes c equal to $4/(4+n)$ where n ranges from 1 to 4. Increasing n increases the resilience against interference and noise. Consequently, the nominal bit rate BR of a LoRa data signal can be calculated with Equation 2.3 [Sem15].

$$BR = f * \frac{b}{2^f} * c \qquad bits/s \tag{2.3}$$

Tuning LoRa modulation parameters has a direct impact on the bit rate, range, and robustness and, hence the transmission time (airtime) and energy consumption of devices. Figure 2.4 shows this impact. Each increase in the SF nearly halves the bit rate and doubles the airtime and energy consumption but enhances the link reliability as it slows the transmission. Also, increasing the

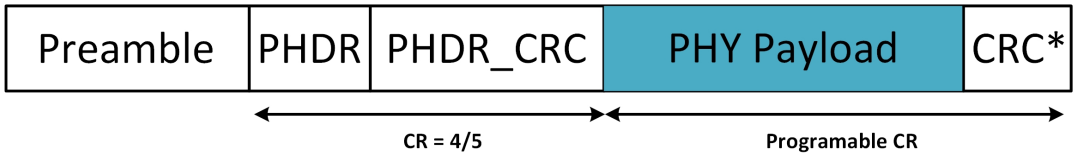


Figure 2.5: LoRa packet Structure (CRC* is only available on uplink packets)

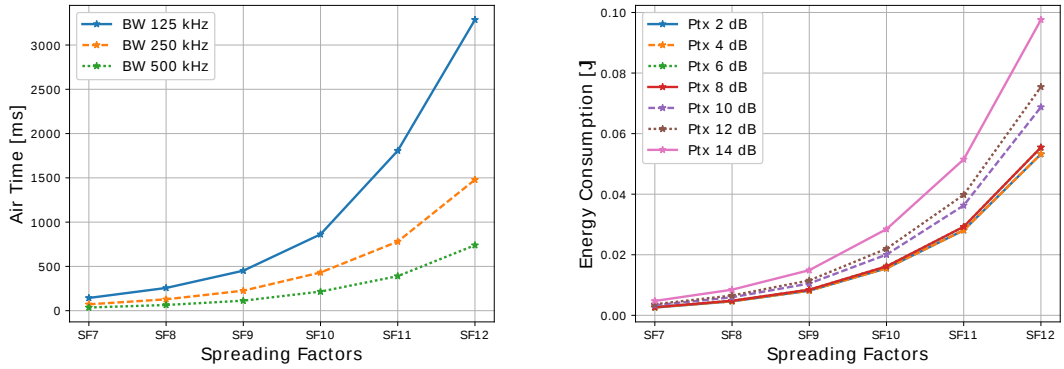
BW increases the bit rate and decreases the airtime and energy consumption but reduces the link reliability as it adds more noise. Considering all the combinations of the modulator, LoRa offers 72 different combinations to transmit LoRa packets. Choosing the transmission parameters is up to each device individually unless a device participates in the Adaptive Data Rate (ADR) mechanism as shown later in this chapter. In this case, the network server controls the transmission parameters of that device. In addition to that, LoRa transceivers allow adjusting the Transmission Power (TP) of the transmitted packets from 2 dBm to 14 dBm in 1 dBm step. Therefore, in total there are 936 different combinations to transmit a LoRa packet.

$$T_p = \overbrace{(k+4.25)\frac{2^f}{b}}^{\text{Preamble Time}} + \overbrace{\left(8+\max\left(0,(c+4)\lceil\frac{8l-4s+28+16-20h}{4(s-2o)}\rceil\right)\right)\frac{2^f}{b}}^{\text{Payload Time}}. \tag{2.4}$$

Figure 2.5 shows the structure of a LoRa packet. It starts with a preamble that is used for synchronization and for defining the modulation format, i.e., the spreading factor used. The preamble is followed by a PHY Header (PHHR) and a Header CRC (PHDR_CRC), whose total length is 20 bits encoded with the most reliable code rate of 4/8. The PHY header contains information such as payload length and whether the optional payload CRC is included or not in the frame.

The airtime of a LoRa packet T_p can be calculated using Equation 2.4 as defined in [Cor13]. The T_p is mainly affected by the symbol time $2^f/b$, which depends on the data rate used. In equation 2.4, the k parameter denotes the number of symbols in the preamble part. The c parameter is the coding rate, ranging from 1 to 4. The l parameter denotes the payload size, whereas h and o are boolean flags, indicating the header presence and the low data rate optimization, respectively.

Figure 2.6a shows the effect of SFs and BWs at code rate CR=4/5 on the airtime



(a) Airtime (b) Energy

Figure 2.6: Impact of SFs and TPs on the Airtime and Energy

Table 2.1: SNR_s thresholds [dB]

SF	SNR
7	-6
8	-9
9	-12
10	-15
11	-17.5
12	-20

to transmit a packet of length 80 bytes. As shown, the fastest combination uses the lowest SF with the highest BW, whereas, the highest SF with the lowest BW achieves the slowest combination and, thus the longest airtime. Figure 2.6b shows the energy consumption for combinations of SFs and TPs at CR=4/5 and BW=500KHz to transmit an 80 bytes packet. As shown, the SF has much higher impact than the TP on the energy consumption, e.g., increasing SF consumes more energy than increasing TP especially for large SFs.

$$\omega = -147 + 10\log(b) + NF + SNR_f \tag{2.5}$$

In addition to that, tuning the SF and the BW influences the link reliability (i.e. sensitivity threshold), which indicates the minimum signal power that a receiver requires to correctly detect the signal. The receiver sensitivity is denoted as ω and can be calculated from Equation 2.5, where the first term is the thermal noise in 1 Hz of bandwidth. The NF parameter denotes the receiver noise figure which is usually fixed for a given LoRa chipset. Finally, SNR_f denotes the signal-to-noise per SF f required for the underlying modulation. Table 2.1 shows these thresholds for all SFs. In this case, receivers can detect

Table 2.2: CIR Thresholds [dBm] [CGM+18a]

Ref. \ Int.	SF_7	SF_8	SF_9	SF_{10}	SF_{11}	SF_{12}
SF_7	1	-8	-9	-9	-9	-9
SF_8	-11	1	-11	-12	-13	-13
SF_9	-15	-13	1	-13	-14	-15
SF_{10}	-19	-18	-17	1	-17	-18
SF_{11}	-22	-22	-21	-20	1	-20
SF_{12}	-25	-25	-25	-24	-23	1

LoRa packets using SF12 and BW125KHz down to -137 dBm.

The SFs of LoRa modulation are forming virtual channels that can enable concurrent transmissions, exploiting the pseudo-orthogonality among the SFs. However, LoRa transmissions are subject to the *capture effect* like other frequency modulation schemes [GG15], which affects the collision behaviour. Specifically, the gateway detects none, one, or both of the collided packets depending on the received powers and the Co-channel Interference Rejection (CIR) thresholds of the SFs. In case the collided packets have the same SF, the gateway detects none if the power difference is less than the intra-SF CIR threshold and detects only the strongest otherwise. Whereas, in case the collided packets have different SFs, both packets can be detected if the power difference is less than the minimum inter-SF CIR threshold or only the strongest otherwise. The CIR thresholds for intra- and inter- SFs are shown in Table 2.2 as reported in [CGM+18a].

2.3.2 LoRaWAN MAC Layer

The LoRaWAN MAC is based on Aloha, where devices transmit immediately whenever the need to (i.e., no Listen-Before-Talk). As LoRaWAN operates in the unlicensed band, its transmissions are subject to the duty cycle and the

Table 2.3: LoRaWAN Regional Parameters for Europe

Data Rates	Configurations	Max App Payloads	Default Channels	Duty Cycle
0	SF12/125KHz	51 bytes		
1	SF11/125KHz	51 bytes	868.10MHz (U/D)	1%
2	SF10/125KHz	51 bytes	868.30MHz (U/D)	1%
3	SF9/125KHz	115 bytes	868.50MHz (U/D)	1%
4	SF8/125KHz	222 bytes	869.525MHz (D)	10%
5	SF7/125KHz	222 bytes		

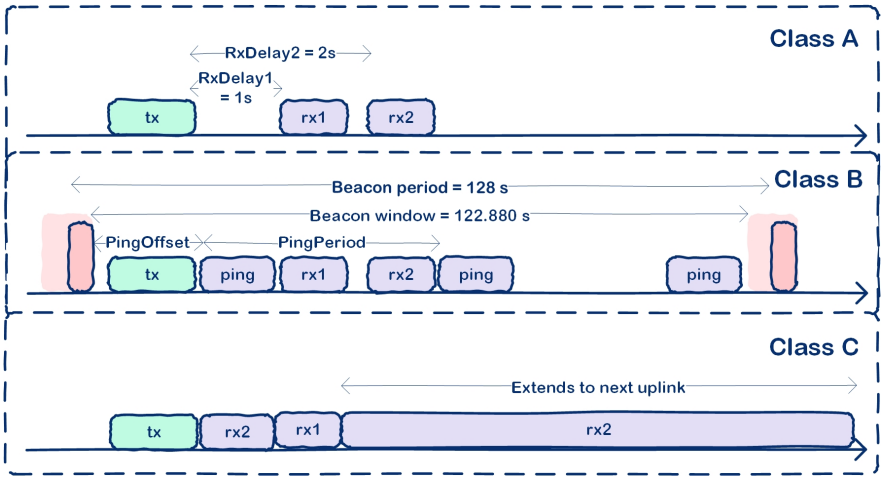


Figure 2.7: LoRaWAN Classes of operations

maximum transmission power regulations of the particular spectrum band. LoRaWAN only uses a subset of the data rates and the power transmission levels that LoRa can support [LoR17b]. The subset is region specific, for example, only 7 data rates and 5 transmission power levels are available in Europe [LoR17a]. Devices operating LoRaWAN have to support at least three channels (868.10, 868.30, and 868.50 MHz) with 1% duty cycle for uplink and downlink transmissions in addition to one channel (869.252 MHz) with 10% duty cycle for downlink transmissions. Table 2.3 shows the regional parameters of LoRaWAN for Europe.

LoRaWAN supports three classes of operation, namely A, B, and C, where each class offers different downlink capabilities to suit a wide range of IoT applications as shown in Figure 2.7. Class A is the mandatory class that all LoRaWAN devices have to support. In this class, each uplink transmission is followed by two receive windows at specific times. Downlink transmissions are only allowed at the beginning of these receive windows. In the first window, the downlink transmission is performed using the same configuration (i.e. data rate and channel) as the previous uplink transmission. While, a fixed configuration (i.e. DR0 (SF12/125KHz) on 869.525 MHz) is used in the second window. On the contrary, class C permits downlink transmissions all the time except when the devices transmit. This is done by extending the second receive window until the next uplink transmission, resulting in high power consumption as devices remain in a receive mode for most of the time.

Class B allows more receive windows than class A but without the high power consumption of class C. Besides the two regular receive windows after each uplink transmission, extra periodic receive windows, called ping slots, are opened

at synchronized times. The synchronization is guaranteed by receiving the gateway beacons that are sent periodically every 128 secs. Before a device can operate in class B, the ping slots periodicity, data rate, and channel used must be made available to the network server. The usable time period between two beacons is called beacon window and it is divided into $2^{12} = 4096$ ping slots of 30 ms each, numbered from 0 to 4095. The ping slot periodicity, *pingPeriod*, of a device is defined to be 0.96×2^p secs, where $0 <= p <= 7$. For a certain periodicity p, the assigned number of ping slots within a beacon window is calculated using 2^{7-p}. In the case of $p = 0$, a device opens 128 ping slots, one slot almost every 1 sec. While in the case of $p = 7$, only one ping slot is opened every 128 secs, which is the maximum supported ping slot period. In order to avoid systematic collisions among nodes operating in class B mode, *pingOffset* is calculated at the beginning of each beacon period to indicate the time of the first ping slot. *pingOffset* is a randomised offset, whose values can range from 0 to $(2^{7-p} - 1)$. The device and the network server use the same randomization function with the same parameters to compute *pingOffset*.

LoRaWAN packet formats are shown in figure 2.8. All LoRaWAN uplink and downlink packets start with single-byte MAC header (MHDR), followed by a MAC payload, and ending with a 4 bytes Message Integrity Code (MIC). The MAC header specifies the packet type, which could be confirmed/ unconfirmed uplink or downlink payload, join-request, or join-response. The MAC payload of the data packet, also called data frame, contains a frame header (FHDR) followed by an optional port field (FPORT) and an optional frame payload field (FRM Payload). The frame header contains the short device address of the device (DevAddr), a frame control byte (FCtrl), a 2 bytes frame counter (FCnt), and up to 15 bytes of frame options (FOpts) used to transport MAC commands.

Devices in LoRaWAN are allowed to use any of the supported data rates (see Table 2.3). However, the network server can be used to adapt and optimize the data rate for each device individually. This is referred to as Adaptive Data Rate (ADR) and when it is enabled the devices will be optimized to use the fastest data rate possible. A LoRaWAN device expresses its interest in using ADR by setting the *ADR* flag (in the FCtrl field) of any uplink MAC header. When ADR is enabled, the network server can control these parameters through the *LinkADRReq* MAC command. Typically, the network server collects the statistics, i.e., RSSIs and SNRs of the most recent packets from all gateways that received these packets. Based on that history, the network server assigns new transmis-

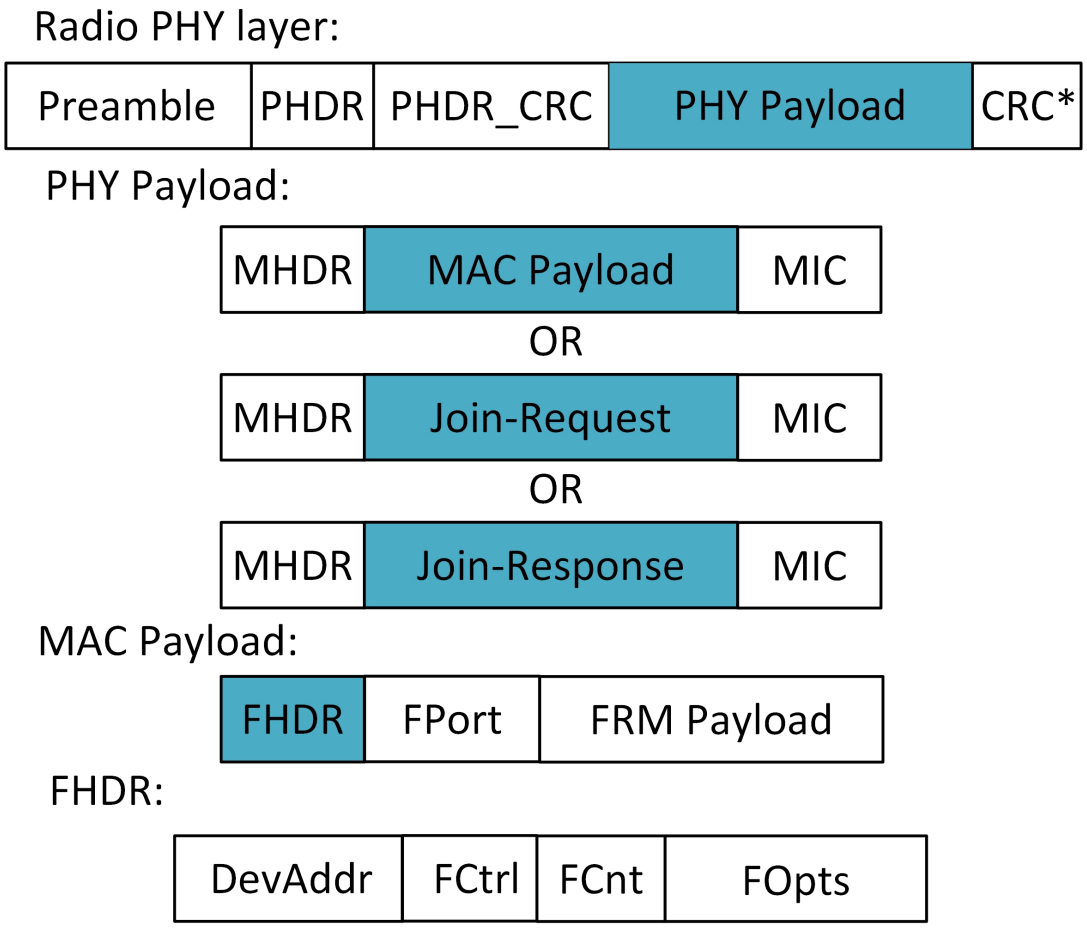


Figure 2.8: LoRaWAN packet structure

sion parameters to prolong the devices' lifetime (the ADR objective) [9]. Once the device receives the new parameters, it responds with a *LinkADRAns* packet as an acknowledgement to the network server and uses the new parameters in the next transmissions.

2.3.3 Simulation Tool

The high number of transmission parameters and potential scenarios in LoRaWAN networks preclude the construction of a large scale test environment [PZ]. Alternatively, computer modelling can be used with the development of a simulation platform. The benefit of having a simulation tool is that it will allow the study of design and deployment options for different LoRaWAN use cases before a costly real deployment. The tool will also facilitate scalability analysis of different designs, which may not easily be feasible in a real environ-

[9] `https://www.thethingsnetwork.org/docs/lorawan/adaptive-data-rate.html`

ment due to the time and cost of deploying a very large number of wireless devices.

LoRaSim [BRVA16] is a custom-build discrete event simulator implement with SimPy [10]. LoRaSim allows to place N LoRa devices and M gateways in a 2-dimensional space. Each LoRa gateway is able to receive multiple LoRa signals (using different SF and BW combinations) for a given carrier frequency. This mimics the behaviour of LoRa gateway chips such as the Semtech SX1301 [11] which can receive 8 concurrent signals as long as these signals are orthogonal. LoRaSim models the communication behaviour of LoRa devices in terms of communication range and collision behaviour (Equations 5 and 11 in [BRVA16]) based on *real-world* experiments that have been performed for all transmission parameter combinations. The communication behaviour is validated based on the XRange SX1272 LoRa module from NetBlocks [12]. This module consists of a low-power STM32L151CC ARM Cortex-M3 microcontroller (32 MHz CPU, 32 kB RAM, 256 kB flash) and a Semtech SX1272 LoRa transceiver. The SX1272 LoRa module is the most commonly employed LoRa chips, which gives credibility to the simulation results.

Although LoRaSim is an adequate simulation tool to assess the communication behaviour of LoRa signals, it is not sufficient for assessing the network performance. The reason is that some simplifications were made e.g. SFs are assumed to be perfectly orthogonal, duty cycle and downlink availability are not implemented and only unconfirmable transmissions can be simulated. These simplifications are not sufficient to model the behaviour of a LoRaWAN network.

Therefore, a new simulation tool was developed. The new tool has been built upon LoRaSim but provides more comprehensive features to accurately simulate a LoRaWAN network. The new tool considers a theoretic packet error model (Section 5.3.1), the imperfect orthogonality of spreading factors (Table 2.2 [CGM+18a]), and the duty cycle limitation (Section 2.2). In addition to that, the new tool supports bidirectional communication by adding the downlink capability (based on LoRaWAN class A devices (Figure 2.7) and a retransmission strategy (up to 8 times before dropping) in case of confirmable uplink transmissions. All added features are required for a proper evaluation of LoRaWAN networks, making the new tool beneficial to the research community.

[10] https://simpy.readthedocs.io/en/latest/
[11] https://www.semtech.com/products/wireless-rf/lora-gateways/sx1301
[12] https://www.netblocks.eu/

Extra features are added to the new simulation tool based on the study that the tool has been used in and also it is given a new name. For example, the tool is called FADR [13] in chapter 4 and focuses on the LoRaWAN ADR algorithm. However, it is called LoRaFREE [14] in chapter 5 and studies the data collection applications, and FUOTASim [15] in chapter 6 and simulates the FUOTA process in LoRaWAN. It is worth mentioning that all these tools are built upon the same core simulator and the extra features are stated in the each chapter.

2.4 Research Gaps

This section summarizes the research gaps identified in the LoRaWAN technology, which stimulated the research in the following chapters.

- **Lack of Interoperability**: LoRaWAN (same as all the unlicensed LPWAN technologies) relies on a simple network stack (see Figure 2.3), where applications can be built directly on top of the MAC layer. This approach complicates the interoperability as it does not support IP protocol (i.e., the principal communications protocol in the Internet protocol suite). Chapter 3 studied this issue and proposed a new IP architecture to the LPWAN technologies. In addition to that section 3.1 reviewed related work in this research area.
- **Fairness in LoRa Modulation**: LoRa modulation supports multiple data rates and its transmissions experience capture effect. These features are expected to create unfairness in data delivery among LoRaWAN devices. This issue has been studied in Chapter 4 and a novel algorithm was proposed to improve the fairness in LoRaWAN. Moreover, section 4.1 reviewed all published related work to this study.
- **Simple MAC Protocol**: LoRaWAN relies on the Aloha protocol, which is known for its poor scalability, which negatively impacts the network reliability. Chapter 5 studied this problem and proposed a novel time-slotted MAC approach to overcome the scalability and reliability issues. In addition to that Section 5.1 reviewed all similar published work to this study.
- **No Support for FUOTA**: LoRaWAN (same as all the unlicensed LPWAN

[13] https://github.com/kqorany/FADR
[14] https://github.com/kqorany/FREE
[15] https://github.com/kqorany/FUOTASim

technologies) does not support FUOTA although it is a critical feature to any wireless deployment. Chapter 6 studied this issue and proposed a new framework to make the FUOTA possible on top of LoRaWAN. Also, section 6.1 reviewed all relevant FUOTA frameworks that have been used for various wireless networks.

Chapter 3

IP Interoperability for LPWANs

To realize the IoT vision of seamless integration with the regular Internet, we have witnessed a trend to support a unified network stack [PAV+12]. This unified network stack, often called *native IoT stack*, consists of the following protocols, e.g. IPv6 at the network layer, UDP at the transport layer, and CoAP at the application layer. The reason behind this is to ensure interoperable operations, which provides open networking, interconnection, and cooperation regardless the Wireless Sensor Networks (WSNs) that devices are connected to. However, most of WSNs are limited in terms of packet size, which cannot support native IPv6 functions, e.g., IPv6 header is 40 bytes. Therefore, 6LoWPAN was proposed as an adaptation layer [1] to compress IPv6 and UDP headers to make their packets suitable for constrained WSNs with low data rate wireless links. While primarily developed for use with IEEE802.15.4-based wireless communication technologies, 6LoWPAN has also been adopted for other wireless technologies such as Near Field Communication (NFC) and BLE. NFC and BLE technologies are quite similar to IEEE802.15.4 in terms of their packet sizes [GPBC17].

LPWANs typically offer packet sizes and data rates up to several orders of magnitude lower than what the 6LoWPAN/6lo [2] usually offer [Far18]. Additionally, most LPWANs do not support fragmentation and typically are designed for applications with uplink-dominant traffic. Most of LPWAN technologies also operate in the unlicensed sub-1GHz band that comes with spectrum regulations in terms of duty cycle and power transmission levels, resulting in very low packet rates. Chapter 2 details the limitations of LPWANs. Because of these limitations, LPWANs are considered as a separate class of wireless networks, which

[1] 6LoWPAN lies between the network and the transportation layers in the OSI model
[2] IPv6 over constrained node networks

are more constrained than the IEEE802.15.4-based networks. Consequently, 6LoWPAN is unsuitable for LPWANs that require a level of adaptation beyond what is used in 6LoWPAN/6lo technologies.

To address this issue, a new IETF working group, the lpwan group, was formed in late 2016 to investigate the adaptation of the IoT stack over LPWANs. The lpwan working group has proposed the Static Context Header Compression (SCHC) [MTG+20] mechanism as a framework to compress/decompress IPv6, UDP, and CoAP headers and handle the fragmentation/reassembling if required. In order to provide efficient header compression, SCHC exploits the characteristics of the LPWANs such as the single-hop topology, where the direction of data flow is only either directly from or to gateways. Additionally, traffic flows are *mostly* known in advance because they are pre-programmed into embedded applications. Therefore, SCHC is based on a *shared static* context that does not change over time and, thus avoids complex synchronization, which is typically the most complex operation in header compression. Consequently, SCHC can compress the headers of IPv6, UDP, and CoAP down to a few bits by omitting known and redundant information and, thus reducing the network overhead, resulting in efficient power operations.

This chapter provides a detailed overview of the SCHC mechanism in Section 3.2. Next, the proposed enhancement of SCHC, the layered SCHC (LSCHC) is presented in Section 3.3. LSCHC is a layered context that saves memory in constrained devices, reduces the processing complexity, and adds flexibility when compressing data flows. In Section 3.4, a novel technique, called dummy mapping, which provides a dynamic context within SCHC is presented, giving more flexibility to SCHC to compress/decompress data flows that are not known in advance. This can improve the overall compression efficiency of SCHC. Performance evaluations of LSCHC and dummy mapping are presented in Sections 3.5 and 3.6, respectively.

3.1 Related Work

Generally speaking, header compression schemes can be divided into three categories: *stateless*, *stateful*, or *hybrid*. Stateless schemes are simple encoding rules, where the encoding does not depend on a data flow but default values are assumed instead. The consequence is that stateless schemes do not achieve good compression ratios when dealing with multiple flows. In contrast, stateful

schemes build a context for each data flow to be compressed, achieving better compression ratios when dealing with multiple data flows simultaneously. However, this comes at a price. Composing the contexts requires sufficient memory and a synchronization mechanism to maintain the contexts. This results in processing time and bandwidth overheads. Also, the condition of the communication medium is critical for the stateful schemes, where a lossy medium can cause a de-synchronization between the compression and the decompression sides. This leads to additional delay in building and recovering contexts, thus adding delay to compressing/decompressing packets. In hybrid schemes, the stateless and the stateful methods are combined into one scheme. By default it operates as a stateless. In case the compression ratio is low, it switches to stateful compression.

RFC4944 [MHCK07] defined the 6LOWPAN_HC1 and the 6LOWPAN_HC2 schemes as stateless header compression protocols for the IPv6 header and the next header (i.e. UDP or TCP headers), respectively, over IEEE802.15.4-based technologies. HC1 assumes default values for the IP version, traffic class, and flow label fields and thus the compressor does not send these fields to the decompressor. The next header field is compressed down to two bits and the hop limit value is carried inline as is. HC1 also does not send the payload length field as it can be inferred from the header of the MAC layer. For the source address and the destination address fields, HC1 can only deal with unicast addresses and only sending the second half of the destination addresses because the source address and the first half of the destination address (network Interface IDentifier (IID)) can be inferred from the MAC addresses. HC1 is an effective scheme for unicast link-local communications but has a very limited effect on global and multicast addresses. Therefore, HC1 is commonly used for local protocol interactions such as IPv6 neighbour discovery or routing protocols. In the best case, HC1 can compress the IPv6 header down to two bytes (one byte for the HC1 encoding and one byte for the hop limit) in the case of unicast link-local communication. However, when the destination address is a multicast address or a global address, the HC1 requires the full 128-bit destination address to be carried in-line.

HC1 extends support for compressing the next header by using HC2, but only for UDP, TCP, and ICMPv6. However, RFC4944 describes only how the UDP header can be compressed, where the UDP length field can be inferred from the header of the lower layer. The commonly used port numbers in the range from F0B0 to F0BF can be compressed down to four bits. The UDP checksum is

carried in-line. HC2 assumes the IPv6 and UDP headers are contiguous headers. Therefore, HC2 cannot compress the UDP header in case an IPv6 extension header is present. In the best case, HC2 can compress the UDP header down to four bytes.

6LOWPAN_HC1 and 6LOWPAN_HC2 are insufficient for most practical uses of IPv6 in 6LoWPAN[Bor14]. In order to address this shortcoming, RFC6282 [TH11] defines the encoding formats, LOWPAN_IPHC and LOWPAN_NHC, to compress/decompress the IPv6 header and the next header, respectively and to overcome the drawbacks of the previous schemes. IPHC is an example of a hybrid header compression scheme that employs stateless compression for the link-local address using 13 bits encoding and employs stateful compression when necessary. In order to compress the global and multicast addresses, IPHC uses an additional 8 bits to store shared contexts for arbitrary prefixes. The context of IPHC allows up to sixteen network prefixes to be compressed when communicating with external networks, however, RFC6282 does not specify any way to build or maintain this context. In the best case, the IPHC can compress the IPv6 header down to two bytes (dispatch and encoding) for link-local communication and three bytes (two bytes for dispatch and encoding, and one byte for stateful context) for multicast and global communications.

IPHC supports compression of the next header using NHC. NHC can compress any arbitrary next header, however, the standard in RFC6282 covers UDP and some of the IPv6 extension headers only. NHC assumes that the UDP length field can be inferred from the lower layer and the checksum field can be recalculated and thus the compressor does not have to send these two fields. In the best case, the NHC can compress the UDP header down to one byte. However, for most practical cases, it compresses the UDP header down to five bytes (one byte for the encoding and four bytes for the ports).

RObust Header Compression (ROHC) [SPL10] is a generic/versatile header compression scheme that can work on different headers such as IP, UDP, TCP, and RTP. ROHC is a stateful scheme in which the compressor and the decompressor share a context. To build the context, ROHC assumes the packets are classified firstly into data flows before being compressed, thus the ROHC takes advantage of the information redundancy for packets belonging to the same flow, where the static redundant information such as source address and destination address etc. are transmitted in the first packet only. Variable information such as identifiers, sequence numbers, etc. are sent in a compressed

form to save bandwidth. Once a packet is classified as belonging to a data flow, the compression is performed according to a profile. A profile defines the compressing function for the different fields in the network headers. Several compression profiles for different data flows, for example, IP only, IPv6/UDP, IP/UDP/RTP, IP/TCP, etc. have been defined.

To ensure context synchronization, ROHC has three modes of operation: Unidirectional mode (U-mode), bidirectional Optimistic mode (O-mode), and bidirectional Reliable mode (R-mode). The U-mode specifies compression over a unidirectional link in which the packets are sent from the compressor to the decompressor. In order to handle the potential errors, the compressor sends periodic updates of the flow context to the decompressor. The O-mode is similar to the U-mode, but operates over a bidirectional link. The O-mode uses a feedback channel to send optional recovery requests and acknowledgements of significant context updates from the decompressor to the compressor. R-mode uses the feedback channel extensively and depends on a strict logic that ensures loss-free context synchronization. ROHC can compress headers of different flows very effectively. In the best case, the headers of a data flow can be compressed down to one or two bytes.

LPWANs are highly constrained networks with very limited packet sizes, in some cases, only one or two bytes are available to transmit all the headers. The aforementioned schemes such as HC1/HC2 or IPHC/NHC cannot achieve the required level of compression for LPWANs. Furthermore, they do not consider the application layer header, e.g., CoAP. Therefore, a new header compression scheme is needed that must be able to compress the application layer header along with the lower layer headers. Although the ROHC may provide the required level of compression, the learning and the synchronization introduce communication overheads that are prohibitive for LPWANs. Furthermore, ROHC entails a significant amount of implementation complexity, which translates directly to an increased amount of processor and memory utilization on devices.

3.2 SCHC in LPWANs Architecture

The SCHC [MTG+20] architecture, as shown in Figure 3.1, provides open bidirectional compression functionality between end-devices and the Internet. The connectivity is possible by compressing the headers of the packets before

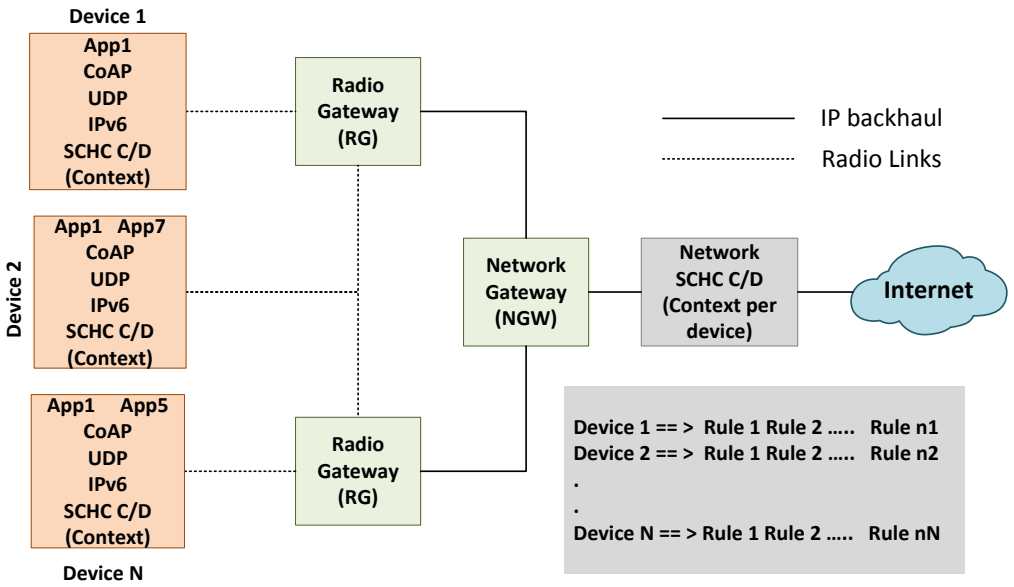


Figure 3.1: The Architecture of LPWANs with SCHC

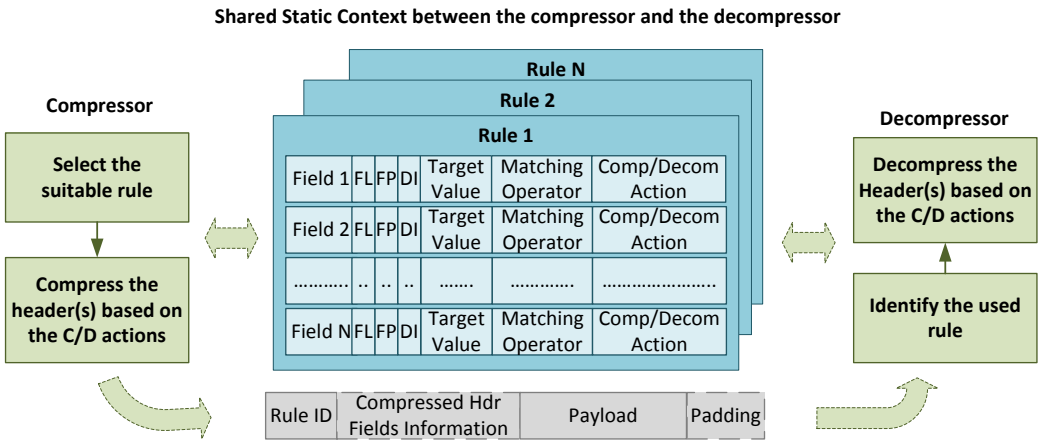


Figure 3.2: SCHC Framework and SCHC Context

transmitting them in the constrained links of LPWAN technologies. The Compression/Decompression (C/D) process of SCHC is performed between the network SCHC C/D unit and each device individually so no major issues could happen becasse of scalability. SCHC is an example of a stateful header compression technique. SCHC relies on a *shared* context between each device and the network SCHC C/D unit to compress/decompress the headers as shown in Figure 3.2. SCHC is based on the fact that the traffic flows are mostly known in

Table 3.1: SCHC Rule for an IPv6/UDP Data Flow

Field	FL	FP	DI	Value	Match Opera.	Comp Decmp Action
IPv6 Version	4	1	Bi	6	ignore	not-sent
IPv6 DiffServ	8	1	Bi	0	equal	not-sent
IPv6 Flow Label	20	1	Bi	0	equal	not-sent
IPv6 Length	16	1	Bi		ignore	compute-*
IPv6 Next Header	8	1	Bi	17	equal	not-sent
IPv6 Hop Limit	8	1	Bi	255	ignore	not-sent
IPv6 DevPrefix	64	1	Bi	FE80::/64	equal	not-sent
IPv6 DevIID	64	1	Bi		ignore	DevIID
IPv6 AppPrefix	64	1	Bi	FE80::/64	equal	not-sent
IPv6 AppIID	64	1	Bi	::1	equal	not-sent
UDP DevPort	16	1	Bi	123	equal	not-sent
UDP AppPort	16	1	Bi	124	equal	not-sent
UDP Length	16	1	Bi		ignore	compute-*
UDP checksum	16	1	Bi		ignore	compute-*

advance in LPWANs since devices run fixed built-in applications. Therefore, all contexts used are *static*.

Static context means that the content of the context does not change over time, avoiding complex re-synchronization, which is the most resource-consuming operation in other stateful header compression schemes such as ROHC [SPL10]. The network SCHC C/D unit manages the contexts of all connected devices of a LPWAN, maintaining a context for each device. A context consists of rules that are lists of field descriptions. Rules can represent compression context for combinations of protocol headers (e.g. a rule for IPv6 header, IPv6/UDP/CoAP headers, or IPv6/ICMP headers). Table 3.1 hows an example of a rule that targets a specific IPv6/UDP flow. Each rule is identified using a rule ID that is specific to each device, where two devices may use the same rule ID for different rules.

A field description in a rule is a tuple containing identifier, value, matching operator and actions to be applied to a certain field of a header.

- ***Field Identifier*** is a unique value to identify the protocol and field a field description applies to.
- ***Field Length (FL)*** indicates the length of the original packet header field.
- ***Field Position (FP)*** indicates which occurrence is targeted in case several the same field is expected to appear multiple times in the header.

- ***Direction Indicator (DI)*** specifies the traffic direction for which the field should be considered (uplink, downlink or bidirectional).
- ***Target Value (TV)*** is the saved value in the rule to be compare with the packet header field value. The target value can be any type and any data structure.
- ***Matching Operator (MO)*** is a boolean function used in the compression only to check the matching between the target value and the header field value. The function returns true in case of matching and false otherwise.
- ***C/D Action*** is a function that describes the process of compression and decompression of the packet header field.

The Internet drafts in [MTG+20] and [MTA20] define a set of matching operators and C/D actions that can be used with the different packet header fields of the IPv6/UDPand the CoAP headers. The matching operators and C/D actions are usually correlated, where a C/D action is generally used with a specific matching operator. For example, *Not-sent* C/A action is generally used with the *Equal* matching operator. In this case, the compressor does not send anything corresponding to this packet header field on which compression is applied. On the other side, the decompressor can use the stored value in the rule to restore the original header value.

The processes of compression and decompression follow different steps as shown in Figure 3.2. The compressor starts by identifying a suitable rule to compress the packet headers. This is done by iterating over all the rules one by one. When the compression is performed in the network SCHC C/D unit, the field directions should be *Down/Both*. In contrast, the direction should be *Up/Both* when the compression is performed in the device. If the direction is not matched, then the rule is not used and the compressor checks the next rule. Subsequently, each header field value is compared to the corresponding target value using the corresponding matching operator. If all the fields of the packet headers satisfy the corresponding matching operators of a rule, the packet header fields are then processed according to the corresponding C/D actions and a compressed header is obtained. Otherwise, the compressor checks the next rule and so on. In case there are multiple rules matching the packet headers, the one that produces the least overhead will be used. The compressor then sends the rule ID followed by the information resulting from the compression, directly followed by the payload and finally padding may be added if the

Rule One

Field	FL	FP	DI	Value	Match Opera.	Comp Decmp Action
IPv6 Version	4	1	Bi	6	ignore	not-sent
IPv6 DiffServ	8	1	Bi	0	equal	not-sent
IPv6 Flow Label	20	1	Bi	0	equal	not-sent
IPv6 Length	16	1	Bi		ignore	compute-*
IPv6 Next Header	8	1	Bi	17	equal	not-sent
IPv6 Hop Limit	8	1	Bi	255	ignore	not-sent
IPv6 DevPrefix	64	1	Bi	FE80::/64	equal	not-sent
IPv6 DevIID	64	1	Bi		ignore	DevIID
IPv6 AppPrefix	64	1	Bi	FE80::/64	equal	not-sent
IPv6 AppIID	64	1	Bi	::1	equal	not-sent
UDP DevPort	16	1	Bi	6457	equal	not-sent
UDP AppPort	16	1	Bi	5686	equal	not-sent
UDP Length	16	1	Bi		ignore	compute-*
UDP checksum	16	1	Bi		ignore	compute-*

Rule Two

Field	FL	FP	DI	Value	Match Opera.	Comp Decmp Action
IPv6 Version	4	1	Bi	6	ignore	not-sent
IPv6 DiffServ	8	1	Bi	0	equal	not-sent
IPv6 Flow Label	20	1	Bi	0	equal	not-sent
IPv6 Length	16	1	Bi		ignore	compute-*
IPv6 Next Header	8	1	Bi	17	equal	not-sent
IPv6 Hop Limit	8	1	Bi	255	ignore	not-sent
IPv6 DevPrefix	64	1	Bi	FE80::/64	equal	not-sent
IPv6 DevIID	64	1	Bi		ignore	DevIID
IPv6 AppPrefix	64	1	Bi	FE80::/64	equal	not-sent
IPv6 AppIID	64	1	Bi	::1	equal	not-sent
UDP DevPort	16	1	Bi	123	equal	not-sent
UDP AppPort	16	1	Bi	124	equal	not-sent
UDP Length	16	1	Bi		ignore	compute-*
UDP checksum	16	1	Bi		ignore	compute-*

Figure 3.3: Two rules have the same IPv6 header in SCHC

datagram is not a multiple of 8 bits. The size of the rule ID varies based on the number of the supporting data flows. In the best case, SCHC can compress the packet headers down to a few bits that equal to the size of the rule ID. At the destination side, the decompressor firstly identifies the rule used in compression using the sent rule ID. Then, it applies the C/D actions to reconstruct the original headers.

3.3 Layered SCHC

SCHC uses a single static context to save the different rules, and rules can cover several layers of the network stack. However, this is not the most efficient method to represent the rules and the method is likely to increase memory usage in the constrained devices. To explain the issue, assume we have two IPv6/UDP flows to the same IPv6 host as shown in Figure 3.3. In this case, the UDP port numbers are the different bits in the two rules. This can happen when the target host runs two concurrent applications on different UDP ports. SCHC would compose a rule for each flow, resulting in a context with two rules that have identical fields for the IPv6 header.

This situation appears in all cases that have two or more data flows sharing the same header(s). This a memory waste that the constrained devices cannot afford. As a result, Layered SCHC (LSCHC), a layered context that consists of multiple contexts is proposed, rather than the single context for the SCHC. Each context in LSCHC consists of rules that target a single protocol stack layer. In this case, there would be a context for the network layer, a context for the transport layer, and a context for the application layer. The proposed context solution is shown in Figure 3.4. To identify rules within their respective layer

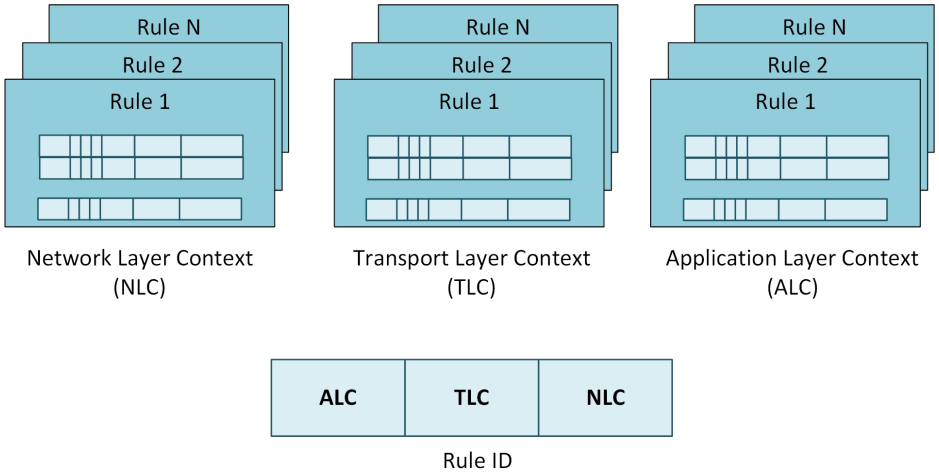


Figure 3.4: The Context structure and the Rule ID of LSCHC

contexts, the rule ID is divided into segments, each segment responsible for identifying the rule used in each context as shown in Figure 3.4. ALC is the segment for the Application layer context, TLC is the segment for the transport layer context and NLC is the segment for the network context layer. The size of each segment can be set based on the LPWAN technology and the number of rules in each context.

Back to the two data flows that are sharing the same IPv6 header example in Figure 3.3. By splitting the rules between layers, resulting in a single rule in the NLC, covering the shared IPv6 header; a separate rule for each UDP data flow. The rule ID can be constructed as follows. To represent the first rule, the LSCHC rule ID should be ALC=0, TLC=1, and NLC =1. To represent the second rule, the LSCHC rule ID should be ALC=0, TLC=2, and NLC=1. LSCHC can save memory on the constrained nodes by storing a single rule for each flow in each layer. Additionally, LSCHC adds flexibility in selecting a suitable rule at the compression side and reduces the processing complexity at the compressor and the decompressor as will be illustrated later in this chapter.

3.4 Dynamic Context for SCHC

SCHC exploits that the data flows from/to LPWANs are known in advance as these devices are usually pre-loaded with certain applications. Therefore, the static context is effective in this sense. However, the real IoT vision is to openly and seamlessly connect LPWAN devices to the Internet, which means applications or services on LPWAN devices should be accessible by any authenticated

flow originating somewhere on the Internet. In principle, such flows cannot always be known in advance. Such unknown flows could be as simple as pinging an LPWAN device or a CoAP request anywhere anytime from all over the Internet. Furthermore, mobility of an external server could lead to a change in its domain that is not supported by the hard-coded SCHC rules. In these cases, some header fields are impossible to be known in advance (e.g. IP address and port number of the client). These issues limit the effectiveness of SCHC because the fields of the unknown flows are sent inline without compression. If the packet cannot fit in one LPWAN packet, fragmentation will take place, leading to multiple transmissions, which increases the overall energy consumption and delay.

3.4.1 Dummy Mapping Technique

In this section, the dummy mapping approach is presented as a technique to compress/decompress some of the header fields that SCHC cannot compress because either they are not known when composing the SCHC rules or these fields may change over time. Specifically, dummy mapping targets requests from the Internet to LPWAN devices, as well as the corresponding responses. These externally sourced traffic items are the most likely to change over time. However, flows initiated by LPWAN devices should be pre-programmed and can be handled effectively using the static context of SCHC.

The idea behind the dummy mapping is to link the value of a header field to be compressed at the network SCHC C/D with a dummy value. As the device and the network SCHC C/D share the same dummy values, the network SCHC C/D sends the index of the dummy value used instead of the actual header field value. The device then reconstructs the compressed header field of the request using the dummy value and sends back the same index in the response. Next, the network SCHC C/D reconstructs the compressed header field of the response using the actual value and forwards the response to the Internet. The dummy mapping assumes values of some header fields e.g. request identifier are not of particular interest to LPWAN devices. However, the values need to be unique and consistent between request and response. The linking between an actual header field value and a dummy value is valid until a corresponding timer is expired. After the expiration, the same dummy value can be re-linked to a different value of a different flow. This dynamic linking overcomes the static context of SCHC and provides a kind of dynamic context to serve dynamic

data flows without re-synchronization with the device, thus it meets the limited characteristics of LPWANs.

In the context of SCHC, dummy mapping is introduced as new matching operator (*dummy-mapping*) and C/D action that can be used in a field description of an SCHC rule. The target value of the field description contains a fixed list of entities. The number of entities indicates the number of flows that the dummy mapping can compress/decompress simultaneously. Each entity is composed of three parts: a dummy value, followed by an actual value and directly followed by a timer. The dummy values part consists of pre-defined values and their order must be the same between the compressor and the decompressor. The actual values part is used to store the header field values at the network SCHC C/D unit, but this part stays empty at the device. The dummy and the actual values must be the same type and size as for the header field that they are intended to deal with. The dummy values do not change over time, however, the actual and timer values are updated over time.

The matching logic of the dummy mapping is presented in the flowchart in Figure 3.5. The compressor firstly checks the field direction as the matching process follows different steps depending on whether the compression is performed in the network SCHC C/D (downlink) or in the device (uplink). If the compression is being done in the network SCHC C/D, the compressor checks if the value of the header field is already linked to a dummy value by comparing this value to the saved actual values. If the value of this header field exists, the matching operator returns true, which means the dummy mapping is valid for compressing this header field. However, if the value of this header field does not exist, the compressor links this value to the first empty or expired entity when found and the matching operator returns true. Otherwise, the matching operator returns false, which means the dummy mapping cannot be used this time. However if the compression is being done in the device, the value of the header field must equal one of the not yet expired dummy values to return true from the matching operator. Otherwise, the matching operator returns false.

Subsequently, if there is a match, the compression is done based on the C/D action. In case the dummy mapping list has multiple entities, the C/D action should contain *mapping-sent(ttl)* and the compressor sends the index that corresponds to the dummy value used to link with the header field. The length of the index depends on the number of the entities in the dummy mapping list. The Time To Live (TTL) in the C/D action is a parameter that indicates the

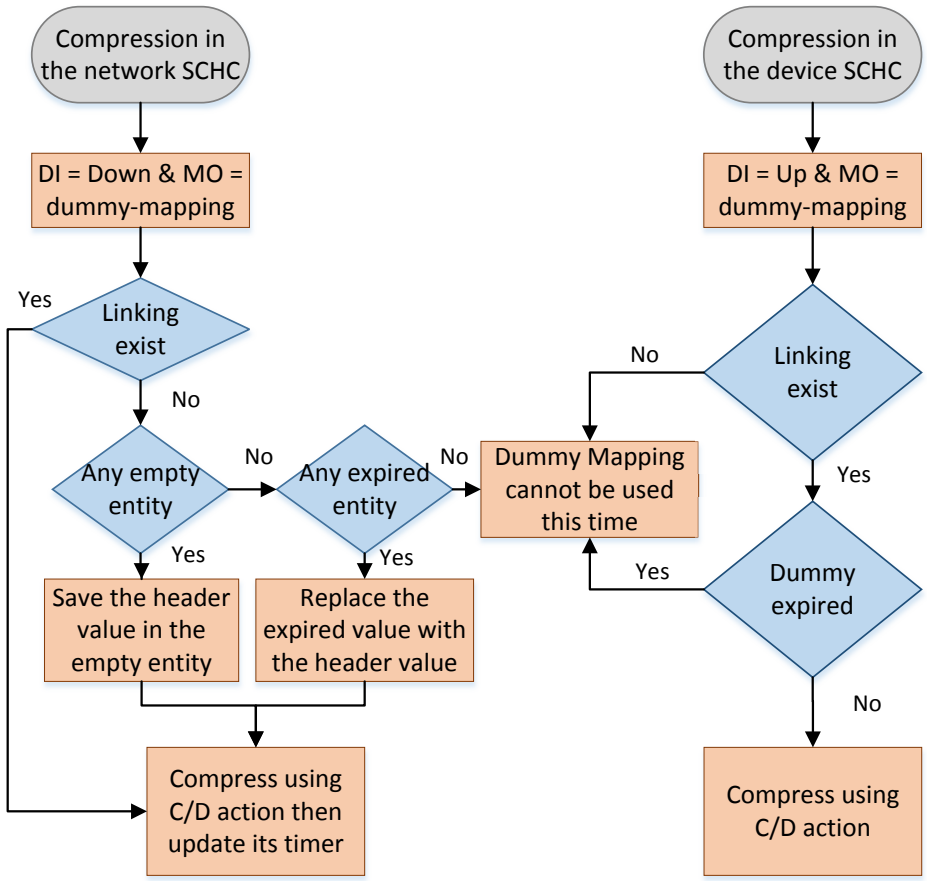


Figure 3.5: The flow chart of the Compression part of the Dummy Mapping

maximum lifetime of mappings. This parameter is used to calculate the timer value in each entity of the dummy mapping list. If there is only one entity in the dummy mapping list, the C/D action should contain *not-sent(ttl)* and the compressor does not send anything to the decompressor. Finally, the compressor updates the timer of the corresponding entity used by adding the TTL value from the C/D action to the current system time when the compression is being done in the network SCHC C/D only.

The decompression performs the steps in reverse order as shown in Figure 3.6. It starts by retrieving the entity used from the list of dummy mappings, which is either identified by the index sent inline (for *mapping-sent(ttl)* action), or the only entry in the list (for *not-sent(ttl)* action). If the decompression is performed in the network SCHC C/D, the decompressor reconstructs the header field using the corresponding actual value. However, the header field is reconstructed

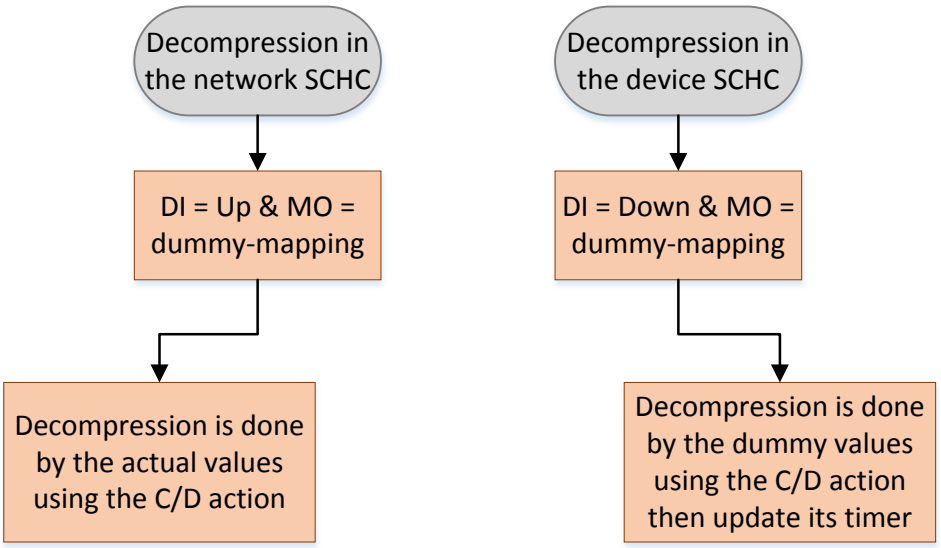


Figure 3.6: Decompression Logic of the Dummy Mapping

using the corresponding dummy value if the decompression is performed in the device. If the decompression is carried out in the device the entity expiration timer is set based on the TTL parameter from the C/D action.

Using the dummy mapping, the linking between the dummy and the actual values is performed dynamically on the network side and without re-synchronization with the devices. That means the devices will not know the actual values. For that reason, the dummy mapping technique is not suitable to handle the header fields whose values must be reconstructed at the devices e.g. destination IPv6 address. Nevertheless, it is suitable to use with the header fields whose values do not matter to the device, but what matters is that these values stay unique for the same flow in order to let the device correctly handle the packets of the same flow. Therefore, suitable header fields that can be compressed/decompressed by the dummy mapping technique are for example the request IPv6 address, request UDP port, tokens and message IDs of CoAP headers.

Table 3.2: A SCHC Rule includes Dummy Mapping for the Scenario in Section 3.4.2

Field	FL	FP	DI	Value	Match Opera.	Comp Decmp Action
IPv6 Version	4	1	Bi	6	ignore	not-sent
IPv6 DiffServ	8	1	Bi	0	equal	not-sent
IPv6 Flow Label	20	1	Bi	0	equal	not-sent
IPv6 Length	16	1	Bi		ignore	compute-*
IPv6 Next Header	8	1	Bi	17	equal	not-sent
IPv6 Hop Limit	8	1	Bi	255	ignore	not-sent
IPv6 DevPrefix	64	1	Bi	FE80::/64	equal	not-sent
IPv6 DevIID	64	1	Bi		ignore	DevIID
IPv6 AppPrefix	64	1	Bi	[[dmy1, ,] ,[dmy2, ,]]	dummy-mapping	mapping-sent(20)
IPv6 APPIID	64	1	Bi	[[dmy1, ,] ,[dmy2, ,]]	dummy-mapping	mapping-sent(20)
UDP DevPort	16	1	Bi	123	equal	not-sent
UDP APPport	16	1	Bi	[[dmy1, ,] ,[dmy2, ,]]	dummy-mapping	mapping-sent(20)
UDP Length	16	1	Bi		ignore	compute-*
UDP CheckSum	16	1	Bi		ignore	compute-*
CoAP Version	2	1	Bi	01	equal	not-sent
CoAP Type	2	1	Dw	CON	equal	not-sent
CoAP Type	2	1	Up	ACK	equal	not-sent
CoAP TKL	4	1	Bi	1	equal	not-sent
CoAP Code	8	1	Dw	0.02	equal	not-sent
CoAP Code	8	1	Up	0.00	equal	not-sent
CoAP MID	16	1	Bi	[[dmy1, ,] ,[dmy2, ,]]	dummy-mapping	mapping-sent(20)
CoAP UriPath	16	1	Dw	/r	equal	not-sent

3.4.2 Scenario: External CoAP POST Requests with Piggybacked Responses

In order to explain the operation of the dummy mapping technique, let us consider a scenario where the LPWAN device implements a CoAP server. The network SCHC C/D unit receives CoAP POST requests from outside clients for a specific resource in the LPWAN device. Typically, the LPWAN device should immediately acknowledge these requests, where the CoAP message ID of the responses should be identical with the message ID of the requests. This is the typical scenario when an external CoAP/HTTP client sends periodic/sporadic control requests to an actuator application within a LPWAN device.

The SCHC rule in Table 3.2 is composed to target the IPv6, UDP and CoAP

headers of the described scenario. The rule should be saved in both the device and the network SCHC C/D to handle the requests and their responses. For this example, the global IPv6 prefix of the LPWAN device address (*IPv6 DevPrefix*) is $FE80 :: /64$ and its identifier part (*IPv6 DevIID*) can be inferred from the Layer 2 address. CoAP in the device is running on UDP port 123 and the path of the target resource is $/r$. The CoAP types and codes of the request and the response messages are confirmable (0.02) and acknowledgment (0.00) respectively. Additionally, the lengths and the checksums fields of all headers are omitted because the decompressor can recalculate them using the lower layers. The hop limit field in the downlink direction is sent inline, but in the uplink direction is omitted because it is known. Therefore, all the known header fields are saved in the rule, where nothing is sent from the compressor and the decompressor uses the saved values to reconstruct the compressed headers.

The dummy mapping in this scenario is used to handle the prefix part of the client address (*IPv6 AppPrefix*) and its identifier part (*IPv6 AppIID*) from IPv6 header, the UDP client port field from the UDP header, and the message ID field from the CoAP header. The values of these fields are likely to change across data flows, except CoAP message ID which does not change over the request/response messages within the same flow. Additionally, these values are not important for the device, but the values must stay unique in the device and must match the response with the request. Thus, these fields are ideal to be compressed using the dummy mapping technique. The rule defines 2 entities per dummy mapping list, where the dummy values per list must be unique. The compressor sends one bit per each dummy mapping list to identify which entity in each list is used in the compression. The TTL value in this scenario is set to 20 seconds. Therefore, with the dummy mapping, the rule can handle client CoAP POST requests from anywhere on the Internet, unlike the case without the dummy mapping in which the values must be sent inline.

3.5 Performance Evaluation of the SCHC/LSCHC

In order to evaluate the performance of SCHC, the protocol is implemented in the Contiki-3.0 [3] operating system [4]. Although Contiki mainly targets IEEE802.15.4-based wireless nodes, Contiki provides an adequate tool to test the performance of SCHC. Firstly, because SCHC depends on a static context

[3] `Contiki-os.org`
[4] `https://gitlab.com/kqorany/SCHC`

that does not require synchronization between the network elements, thus, the technology and condition of the channel do not influence its behavior. Furthermore, Contiki provides a good framework to work on and test the SCHC against the header compression techniques currently implemented in Contiki, which are IPHC for IPv6 and NHC for the next headers. Finally, there is a move to expand Contiki to support LPWAN technologies [5].

For implementation purposes, a dispatch identifier is defined for the SCHC (3 bits) to be compatible with 6LoWPAN. The size of the rule ID is set to 5 bits, which allows a device to handle 31 different rules (although this value could be increased if needed). As recommended by the SCHC draft [MTG+20], the Concise Binary Object Representation (CBOR) [BH13] was used to represent the target value, matching operator, and C/D actions of the SCHC fields. With small modifications, the generic CBOR was ported from the RIoT operating system to Contiki.

The topology setup consists of two TmoteSky devices in which one of them acts as a sender and at the same time is the root of the Routing Protocol for Lossy Networks (RPL) Destination Oriented Direct Acyclic Graph (DODAG), and the other device acts as a receiver. This one-hop topology is similar to the star-topology of LPWANs in which all nodes communicate through a gateway. The ipv6/rpl-udp example was used from the Contiki examples repository in which the sender periodically, every minute, sends "Hello" messages to the receiver. Small packet sizes (less than 20 bytes) were used to avoid the effect of packet fragmentation. The ipv6/rpl-udp example produces three different flows between the sender and the receiver. Firstly, an IPv6/ICMPv6 flow with unicast link-local address at the source and multicast with link-local scope address at the destination; this flow is used to discover the neighbors at the beginning of communication. Secondly, an IPv6/ICMPv6 flow with unicast link-local addresses at the source and the destination; this flow is used to create and maintain the RPL DODAG. Thirdly, an IPv6/UDP flow with global addresses at the source and the destination; this flow is used to transmit the "Hello" messages.

To compress the three flows using SCHC/LSCHC, three rules are composed in which each rule targets a flow. The compression ratio metric is reported, which indicates the ratio between the uncompressed size and the compressed size. Higher compression ratio indicates higher compression efficiency and a lower amount of data to be sent on air.

[5] `github.com/Wi6labs/lorafabian`

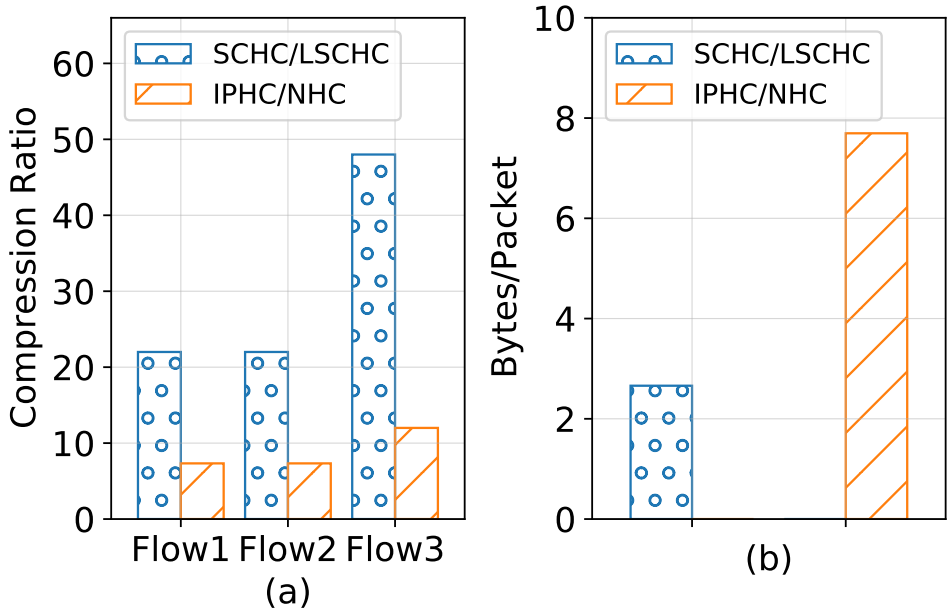


Figure 3.7: compression ratio in the case of SCHC/LSCHC vs IPHC/NHC

Figure 3.7a shows the compression ratio for each data flow in case of SCHC/LSCHC and IPHC/NHC. As shown, SCHC/LSCHC can achieve a three times higher compression ratio than IPHC/NHC in all flows. The first and the second flow are similar (IPv6/ICMPv6). SCHC can compress this flow down to two bytes (one byte for dispatch and rule ID and one byte for ICMPv6). As the logic for compressing the ICMPv6 header is not implemented in Contiki, IPHC/NHC compresses these headers down to six bytes (two bytes for the dispatch and IPHC encoding and four bytes for ICMPv6). Regarding the third flow (IPv6/UDP), SCHC/LSCHC can compress the two headers down to one byte (dispatch and rule ID), however IPHC/NHC compresses the two headers down to four bytes (two bytes for dispatch and IPHC encoding, one byte for the stateful compression, and one byte for UDP).

Figure 3.7b shows the average transmitted bytes, the headers only, per packet after running the emulator for six hours. The proportion of the IPv6/UDP and the IPv6/ICMPv6 packets is the same in SCHC/LSCHC (358.33, 301.66) and IPHC/NHC (350, 308.33). The bytes of the MAC frame and the payload from the calculations were omitted as they are the same in both header compression schemes. As shown, SCHC/LSCHC sends on average 2.66 bytes/packet for the headers, however, IPHC/NHC sends on average 7.69 bytes/packet for the headers.

Figure 3.8 illustrates the total transmission time to send the average flow head-

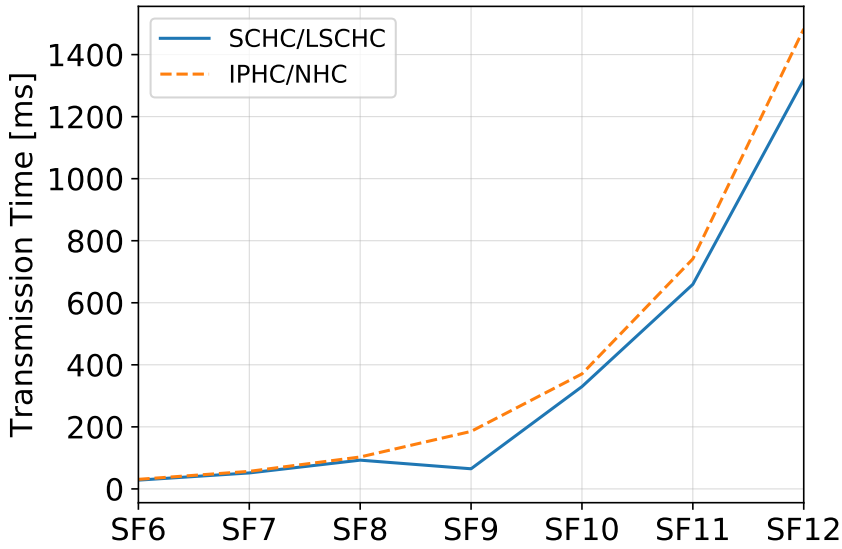


Figure 3.8: Transmission Time in the case of SCHC/LSCHC vs IPHC/NHC

ers over LoRa for different Spreading Factors (SFs) as an example of a LPWAN technology. Higher SF means more range and better reception but also means more transmission time. The calculations are performed for the 125 KHz bandwidth case, 4/5 coding rate, and 8 symbols per preamble. Semtech's LoRa calculator was used to calculate the transmission times [Cor13]. This metric has direct impact on the power consumption of the devices because the power consumption of low power devices depends very much on the transmission time. Therefore, SCHC/LSCHC helps devices to live longer with the same power source compared to IPHC/NHC.

3.5.1 Insights from Implementing SCHC/LSCHC

3.5.1.1 Registering Rules

An end device runs application(s) that produce/accept specific data flows. In order to compress/decompress effectively the headers of packets in these data flows, the network and the device C/D units must share the same rules and their rule IDs must be identical, otherwise, a mistake might happen, causing errors in decompressing the packets. However, it cannot be assumed that the network C/D knows in advance all the flows of all applications that can join the network later. Therefore, some initial synchronization is necessary. The SCHC draft does not specify how this synchronization might be performed. Therefore, the network C/D is suggested to support a layered context with a set of rules that might be suitable for the most common data flows. Subsequently, each device

within an LPWAN should offline pick the most suitable rules for its applications to achieve the maximum compression level. Devices only store a small set of rules that they need and for that the devices need a small rule ID to represent the set. However, the network C/D stores a much larger set of rules (covering all applications and more), and therefore it will use a larger rule ID to represent the large set. This will require a mapping from device address and device short rule ID to the longer, network C/D, rule ID. Two devices may use the same rule at the network C/D, but with different short rule IDs. It is the short rule IDs that will be sent between a device C/D and the network C/D to specify the used rule in compression.

This proposed method will save memory in the network C/D as well by just saving one version of each rule, rather than building a context specific to each device as originally proposed in the SCHC draft. Furthermore, this method saves bandwidth in the constrained networks by just sending the short identifier among the compression and decompression sides. In the case of a new flow that requires a new rule at the network C/D, a manual administration process is required using an unconstrained connection to the network C/D to register the new rule with a unique long ID. This new rule can then be used by different devices. In all cases, there is no requirements for an online learning process between a device C/D and the network C/D, thus saving on the limited bandwidth of the LPWANs.

3.5.1.2 Matching and Selection of Rules

One of the main functions of the C/D unit is to select a suitable rule to compress the header(s). The SCHC draft specified a suitable rule as one where all the rule fields match the packet fields according to the matching operators. If such a rule is found, the packet is processed using the corresponding C/D actions to this rule, otherwise the packet is sent without compression. However, this is not accurate because the decompressor always expects a rule ID in the packet. Therefore, if there is no matching, the compressor side must send a special ID meaning the packet is not compressed. Furthermore, selecting the first matching rule with the header(s) may not be the best matching approach because there could be more than one rule that matches with the header(s). To get the best solution, the compressor should test all available rules and then select the rule that achieves the best compression ratio.

LSCHC, which was presented in subsection 3.3, adds flexibility in selecting the

most suitable rule that matches the header(s). Assume a compressor has a rule that targets IPv6/UDP headers. In the case of SCHC, to use this rule, a matching must occur with all fields in the IPv6 header and the UDP header. However, the compressor may produce/receive packets that match with the IPv6 header only or with the UDP header only. With SCHC, the compressor would not be able to use this rule. With LSCHC, because the context is layered, LSCHC can compress the packets that match only with the IPv6 header or match only with the UDP header using the corresponding rule. Therefore, LSCHC is more flexible and can achieve a higher gain in terms of compression ratio compared to SCHC in this scenario.

3.5.1.3 Processing Rules

Processing the context in the case of SCHC is rule specific, not field specific because rules target multiple headers, thus, the number of fields in a rule is not identical. This means that rule specific processing is not scalable as it requires adding a new processing logic each time a new rule is added. LSCHC solves this problem by separating the layers and processing each layer individually. Isolating the layers makes the processing logic header specific as all headers in a layer are known e.g. IPv6 in the network layer and UDP, ICMPv6 and TCP in the Transport layer, etc. This makes the header specific processing field specific as well. Therefore, processing the context in the case of LSCHC is generic and can work on any kind of rule due to its field specific approach.

3.6 Performance Evaluation of the Dummy Mapping Technique

In this section, the TTL value is calculated. This value must be used in the C/D action and the number of entities in the dummy mapping list to handle an expected request rate. Subsequently, an analytical model is derived to measure the efficiency of the dummy mapping in terms of the compression ratio that can be achieved. Next, a numerical example for a device in a LoRaWAN [LoR17b] is provided.

3.6.1 Analytical Model

The TTL in the dummy mapping C/D action must satisfy Equation 3.1, where RTT_L is the longest round-trip time between the network SCHC C/D and the device. Satisfying Equation 3.1 ensures that the response of any request is received in the network SCHC C/D unit before the corresponding timer of the entity used expires. In perfect channel conditions, the longest round-trip time is defined as the time needed for the longest request T_{Lreq} to be sent plus the longest time needed for the response to be prepared T_{pre} plus the time needed for the longest response T_{Lres} to be received back. Nevertheless, the last definition neglects the effect of the channel on the transmission times. To include that effect, the longest round-trip time should cover the highest possible delay in the channel due to collisions. Therefore, RTT_L in Equation 3.1 considers the maximum number of MAC retransmissions R_{max} in the calculation, which means any transmission needs to transmit R_{max} times before being received. By considering the longest possible delay due to the channel and the application behaviour, a deterministic TTL value can be set per each application per each device.

$$TTL \geq RTT_L \tag{3.1}$$
$$where$$
$$RTT_L = R_{max}(T_{Lreq} + T_{Lres}) + T_{Lpre}$$

In order to understand the behaviour of the dummy mapping compression technique in a device, the dummy mapping list is modeled as a $M/D/N/N$ queue [AR01]. In this queuing model, the requests arrive in the network SCHC C/D according to a Poisson process with rate λ and mean inter-arrival time equal to $1/\lambda$. The service times are deterministic values equal to TTL and service rate $\mu = 1/TTL$. There are N entities in the dummy mapping list to compress the incoming requests. Each newly arriving request is compressed using one of the entities if there is an empty or an expired one, otherwise, the request cannot be compressed using the dummy mapping. Hence, Equation 3.2, which is known as *Erlang's loss formula* [AR01], represents the probability of a new request finding all the entities busy i.e. timer not-expired-yet. Consequently, $(1 - P(\rho, N))$ is the probability that a new request finds at least one free entity to use. Based on the probability $P(\rho, N)$, the ratio of requests that are compressed can be determined for given values of N and λ.

Table 3.3: The rate when the Erlang's loss formula is less than 0.005

N	1	2	3	4	5	6	7	8
max ρ	0.005	0.1054	0.3490	0.7012	1.132	1.622	2.158	2.730

$$P(\rho, N) = \frac{\rho^N/N!}{\sum_{i=1}^{N} \rho^i/i!}, \quad where \quad \rho = \lambda/\mu = \lambda TTL \tag{3.2}$$

$$P(\rho, N) = \frac{\rho P(\rho, N-1)}{N + \rho P(\rho, N-1)} \tag{3.3}$$

Fortunately, it is easy to derive a simple recursion for Equation 3.2[Ang01]. Starting with $P(\rho, 0) = 1$, we can compute $P(\rho, N)$ using Equation 3.3 for different N and ρ values. Consequently, with the desired probability of using the dummy mapping, we can get the minimum number of entities that achieve that probability for a given value of ρ. For example, Table 3.3 shows the maximum ρ values for different number of entities N to ensure that $P(\rho, N) \leq 0.005$.

The compression ratio is defined as the ratio between the size of the uncompressed and compressed traffic. The compression ratio gives an indication of the efficiency of the compression, where a higher ratio indicates a higher efficiency. Therefore, the compression ratio in case of using the dummy mapping CR_{dummy} for a header field with length FL can be calculated with Equation 3.4. Using dummy mapping, header fields are compressed to the index of the corresponding entity used, where a field with length FL is compressed down to $\lceil log_2 N \rceil$ bits.

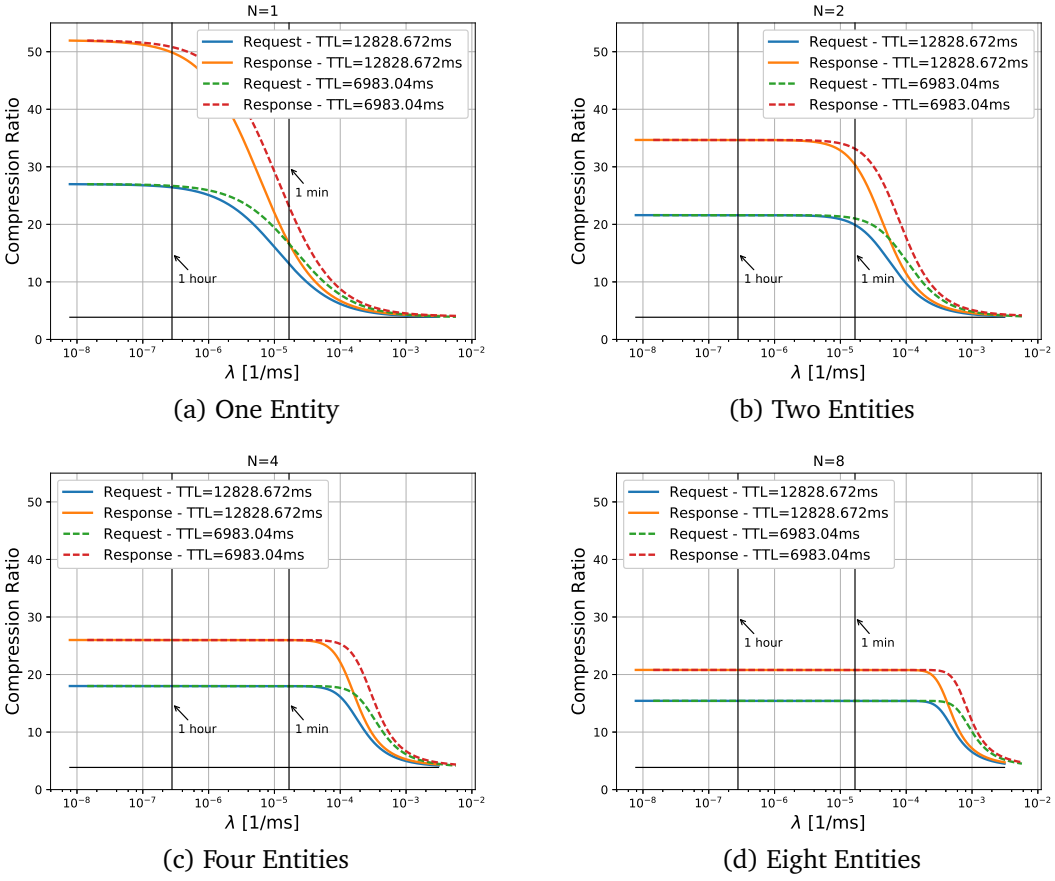
$$CR_{dummy} = \frac{FL}{(1 - P(\rho, N))\lceil log_2 N \rceil + P(\rho, N) FL} \tag{3.4}$$

3.6.2 Numerical Example

In this example, a LoRaWAN device [LoR17a] is considered, running the scenario in Section 3.4.2. The device in this scenario sends the responses immediately after receiving the CoAP POST requests and the rule in Table 3.2 targets the IPv6, UDP and CoAP headers. LoRaWAN supports 7 different bit rates (Table 4 in [LoR17a]) that devices can use to communicate with the gateway and the gateway uses the slowest bit rate to communicate with the devices. Additionally, LoRAWAN defines a maximum MAC packet size of each bit rate (Table 8 in [LoR17a]). In order to set the TTL value, these maximum packet sizes and $R_{max} = 3$ to calculate the RTT_L are used, which considers the worst case. As

Table 3.4: RTT_L values of LoRaWAN bit rates at $R_{max} = 3$

Data Rate	Physical bit rate [bit/s]	Longest Packet [B]	Transmission Time of Longest Packet [ms]	RTT_L [ms]
0	250	51	2138.112	12828.672
1	440	51	1150.976	9867.264
2	980	51	616.448	8263.68
3	1760	115	615.424	8260.608
4	3125	242	666.112	8412.672
5	5470	242	379.136	7551.744
6	11000	242	189.568	6983.04



(a) One Entity

(b) Two Entities

(c) Four Entities

(d) Eight Entities

Figure 3.9: Compression Ratio of the Dummy Mapping

the responses are sent immediately, $T_{Lpre} = 0$ for this scenario. Consequently, RTT_L equals three times the transmission time of the longest frame size using the lowest bit rate T_{Lreq} plus the transmission time of the longest packet size using the device bit rate T_{Lres}. Table 3.4 shows the RTT_L values of LoRaWAN bit rates. From these values, the TTL value can be set for each device to satisfy Equation 3.1. Semtech's LoRa calculator [Cor13] was used to calculate the transmission times.

Without dummy mapping, the compression in this scenario is performed by sending the prefix part of the application address, UDP application port, and the CoAP message ID inline. one byte is used to represent the rule ID as in [ACP17]. Therefore, the compression ratio of the requests and the responses *without the dummy mapping* are 3.85 and 4 respectively. On the other hand, Figure 3.9 shows the compression ratios for the requests and the responses in case the dummy mapping list has one entity (Figure 3.9a), two entities (Figure 3.9b), four entities (Figure 3.9c), and eight entities (Figure 3.9d). All figures show the results of the lowest and the highest $TTL = RTT_L$ values from Table 3.4. In Figure 3.9, the horizontal lines represent the compression ratio without dummy mapping whereas the vertical lines represent the mean inter-arrival time of requests.

Figure 3.9 shows that for each value of N the compression ratio decreases from the highest value to the lowest value as the request rate λ is increased. The highest value for each N happens when $P(\rho, N)$ equals 0 (requests always find an available dummy mapping entity) and as the $P(\rho, N)$ increases (i.e. by increasing the λ) the compression ratio declines to the lowest value. This happens when $P(\rho, N)$ equals 1, which means sending the header fields inline without compression. The highest value shown in Figure 3.9 decreases as the number of dummy mapping entities N increases. This is because the number of bits required to identify the dummy mapping entity also increases, reducing the compression ratio. It should be noted that there is no benefit in using values of N that are not powers of 2 e.g. $N = 3$ because it achieves the same highest compression ratio as $N = 4$, however, the case of $N = 4$ maintains this highest compression ratio for larger request rates. Finally, the results show that lower values of TTL at the same request rate achieve higher compression ratios. This is because reducing TTL increases the probability of finding expired dummy mapping entities, thus the probability of compressing the traffic with the dummy mapping mechanism.

LoRaWAN operates in unlicensed bands, which restricts the duty cycle to low values, e.g. in Europe 1% duty cycle should be respected in the $863 - 870MHz$ band. With this low duty cycle, the expected request rate of LoRaWAN applications is very low with mean inter-arrival times in the order of minutes or even hours. Therefore, the rates stated in Figure 3.9 are more than enough for a wide range of applications suported by LoRaWAN. The compression ratio of the requests and the responses in case of $N = 1$ and $1hour$ mean inter-arrival time of the requests are 27 and 52 respectively. That means, in this scenario, the

dummy mapping provides an increase of 600% and 1200% in compressing the request and the response respectively compared to the regular static compression methods of SCHC. This, in turn, has a significant positive effect on device lifetime.

3.6.3 A Discussion of the Dummy Mapping Technique

The results presented in Section 3.6 show the improvement that the dummy mappings approach can achieve when compared to the static context compression SCHC. With dummy mappings, a static context scheme behaves more like a dynamic context scheme that still considers constant value fields; therefore, dummy mapping does not require re-synchronization. It can be seen that the advantage of the dummy mapping scheme is that the context establishment phase is not necessary. For a dynamic context scheme, this phase requires at least one round-trip exchange performed uncompressed. In many cases, such as the scenario discussed in Section 3.4.2, this exchange is actually the only one performed; therefore, the established context will not be used subsequently. Furthermore, the delay and success of the context establishment phase are affected by channel parameters and conditions such as bit rate, error rate, and congestion. With these taken into account, the effective compression ratio achievable with a dynamic context can be halved [WK03]. These factors do not affect the dummy mappings scheme because the message exchange is always performed compressed. This last statement is conditioned on the availability of dummy mapping entities, which, however, is also a condition for dynamic context schemes that will also suffer from memory constraints. The only disadvantage of the dummy mapping scheme, compared to a dynamic context scheme is that with dummy mappings the compression is lossy. This is discussed next.

3.6.3.1 Impact of lossy compression

First affected is the checksum, which is mandatory in all protocols in layers above the network layer (i.e., IPv6) to verify data integrity between the sender and the receiver. However, if dummy mapping is used, the reconstructed header in the decompressor is different than the original header, so the decompressor calculates different checksum values than the values that were calculated in the compressor. This may lead to the decompressor rejecting all packets that were compressed using the dummy mapping. Nevertheless, in the context of LPWANs, the compressor most probably does not send the checksum values to

save room for the application payload and a layer 2 CRC is used instead to ensure correct transmission. If the checksum value must be explicitly sent and dummy mapping is used, the compressor must then recalculate the checksum using the dummy values instead of the actual ones.

The lossy compression of dummy mappings can introduce security risks if the targeted fields are used for authentication. For example, an access control list would break with dummy mapping compression because two or more distinct flows would use the same dummy identifier. On the other hand, the dummy mapping scheme targets the requests that are external to the network, and that will be application requests above the network layer, such as CoAP. If security is implemented in these requests it will be through transport security such as DTLS, that do not rely on protocol headers but exchange secure information (public key certificates) in the payload. In these cases, dummy mappings will not affect the security. Another potential security issue, from an implementation point of view, is if a secure material such as keys is stored locally and indexed by identifiers from the protocol headers (*e.g.*, IPv6 address). This would lead to an intentional or accidental break of authentication because two or more flows would share the same identifier, thus the same keying material.

3.6.3.2 Impact of TTL value

Considering the absolute worst case round-trip time RTT_L to set the TTL value ensures that the message exchange of CoAP request/response model is completed before the entity expires at the device and the network SCHC C/D unit. This leads to a long TTL value that reduces the highest compression ratio when the request rate increases as shown in Figure 3.9. Although lowering the TTL value maintains that highest compression ratio for higher request rate, it *may* lead to a compression rejection at the device or wrong decompression at the network SCHC C/D. This is because the entity at the device may expire before the device sends the response or the entity at the network SCHC C/D might be relinked before the response is received at the network SCHC C/D. Eventually, this may trigger a retransmission at the application layer, which, compressed or not, results in higher latency and energy consumption. Nevertheless, a lengthy TTL is not a critical issue in the context of LPWAN, where the request rate is expected to be very low. Even in case of a high request rate, the same highest compression ratio at a specific N can still be achieved at the expense of memory by supporting, for example, two exact rules, using dummy mapping list with N

entities, where each rule handles a half of the request rate.

3.6.3.3 Complexity of the dummy mapping

The run-time of the compressing/decompression in case of using the dummy mapping is linear $O(N)$, where N is the number of entities in the dummy mapping list since the logic of compression and decompression as shown in Figure 3.5 and Figure 3.6 respectively iterate over all entities just once. This is an important property because the LPWAN devices are resource constrained and are expected to be deployed with a low power central processing unit. As the expected request rate in the context of LPWANs is very low, the number of entities that are expected to be deployed would be small so as to not create a processing problem.

3.7 Conclusions

SCHC is a header compression mechanism that promises to enable IP-based traffic over very constrained networks such as LPWANs. In this chapter, the performance of SCHC was evaluated against other header compression mechanisms. The results showed that SCHC can achieve higher compression efficiency, resulting in lower transmission time of packets and, thus lower energy consumption. However, composing a large number of SCHC rules might require a lot of memory to maintain them. To address this issue, LSCHC was proposed as a new efficient mechanism for saving compression rules on constraint devices without compromising the compression efficiency of SCHC. Another issue that limits the applicability of SCHC is its static context nature that only handles fixed or pre-programmed data flows. For this, the dummy mapping technique was proposed as a dynamic mechanism to handle unknown data flows, which would help to increase the applicability of SCHC, particularly in the context of the Internet of Things. An analytical model was also derived to measure the performance of the dummy mapping. The results showed that dummy mapping increases the average compression ratio obtained by SCHC by up to 850% (see Figure 3.9a) for hourly LoRaWAN traffic and up to 575% for per minute traffic.

Chapter 4

Fair Resource Management in LoRaWAN

LoRaWAN (chapter 2) is a key technology in the LPWAN category with a strong ecosystem of vendors, operators, developers, and IoT platforms. In this chapter, the PHY (i.e., LoRa modulation) and MAC layers of LoRaWAN technology are studied along with its standard resource allocation algorithm. The study is aimed to figure out the fairness of the technology in terms of data delivery, which is a desired property in any wireless networks to assure a certain QoS [HPON13].

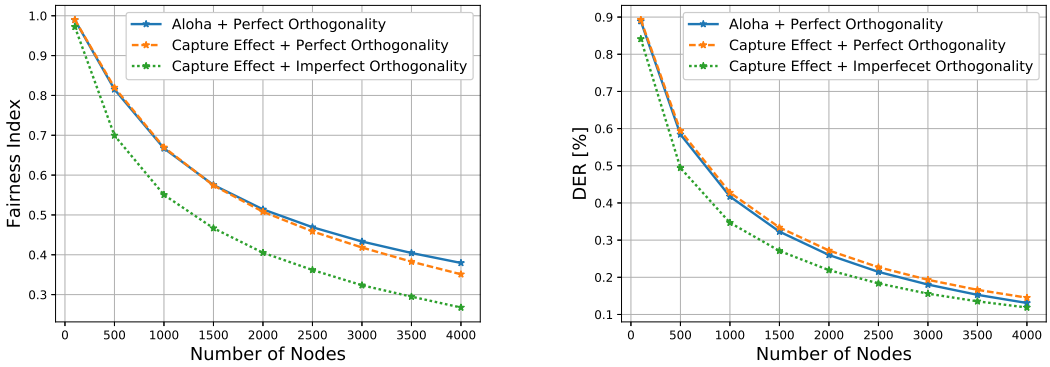
LoRa modulation, similar to other frequency-based modulations, is subject to the *capture effect* [BRVA16]. In this case, depending on the specific radio communication conditions, a LoRaWAN gateway can decode one, all, or none of the colliding packets transmitted by multiple nodes. The decoding depends on the spreading factors and the RSSIs of the collided signals. When two packets, using the same spreading factors, overlap in time, with one packet stronger in terms of the received power than the other by the Co-channel Interference Rejection (CIR) threshold, the stronger packet drowns the weaker. While If the power difference is below the CIR threshold, none of both packets can be decoded correctly. Even when the collided packets use different spreading factors, this effect can still be observed, because the spreading factors used by LoRa modulation are not perfectly orthogonal [CGT+17]. However, if the power difference, in this case, is below the CIR threshold, the two collided packets can be decoded correctly. The value of the CIR threshold varies according to the collided spreading factors. Croce *et. al.* measured the CIR thresholds for all

spreading factor combinations in [CGT+17].

The capture effect is known to cause the near-far problem or hearability problem [She11]. To explain the problem, consider a LoRa/LoRaWAN gateway and two nodes, one close to the gateway and the other is far away. If the two nodes transmit simultaneously and at equal transmission power levels, the gateway will receive more power from the nearer node than the farther node due to transmission attenuation. In this case, because of the capture effect, the gateway will always decode the stronger transmission from the nearer node, making it unfair for the farther node that less likely to see its transmissions decoded correctly.

In addition to that, LoRa modulation supports multiple data rates (combination of SF, BW, and CR) (Equation 2.3). These multiple data rates affect the probability of collision for LoRa/LoRaWAN nodes. To clarify this, consider twenty LoRa/LoRaWAN nodes are distributed around a gateway wherein the first ten use the highest data rate and the other ten use the lowest data rate. Also, consider that all nodes transmit the same packet size at the same rate. In this case, the packet transmission time is shorter for the nodes that use the highest data rate than the other nodes. Indeed, the longer the transmission time the higher the probability of a collision due to increasing the vulnerability time. Consequently, the ten nodes that use the lowest data rate are more likely to experience more collisions, making it unfair for those nodes in terms of data delivery.

LoRaWAN supports a resource allocation algorithm, called Adaptive Data Rate (ADR), which is responsible for optimizing the network performance. Specifically, the ADR allocates the minimum data rate and the minimum transmission power level wherein the RSSIs from each node (regardless of its position from the gateway) are higher than the receiver sensitivity (depending on data rate). Equation 2.5 defines the relationship between receiver sensitivity and the data rate. Using LoRaWAN ADR algorithm, each node does not consume unnecessary power to transmit its packets. However, when it comes to the data delivery for each node this mechanism of allocation is unfair because it does not handle the unfair characteristic of LoRa modulation. For the medium access, LoRaWAN uses Aloha protocol, which is supposed to be a fair MAC protocol. Nevertheless, the above PHY characteristic along with LoRaWAN's ADR algorithm introduce unfairness in the data delivery, favouring transmissions from nodes nearer to the gateway and by those that use high data rates.



(a) Fairness Index (b) DER

Figure 4.1: Study of LoRaWAN fairness

To quantify the impact of these factors on fairness, a simulation is used, where the fairness is measured using Jain's fairness index [JCH84]. Jain's fairness index, known also as the fairness index, is the most used quantitative fairness measure in the allocation problem [HPON13]. For the data delivery, the fairness index is defined as,

$$\zeta = \frac{(\sum_{i=1}^{n} DER_i)^2}{n \sum_{i=1}^{n} DER_i^2}, \tag{4.1}$$

where $0 \geq \zeta \leq 1$. DER_i denotes the Data Extraction Rate of the i^{th} node and n denotes the total number of nodes. DER_i is defined as the ratio of the received messages by at least one gateway to the transmitted ones by the i^{th} node over the simulation time. A higher ζ value represents fairer resource allocation from the system perspective, happening when the majority of the nodes get roughly the same DER.

In the simulation [1], n nodes are uniformly distributed around a LoRaWAN gateway. Figure 4.1a shows the fairness index in three different cases: Aloha + perfect orthogonality (case 1), capture effect + perfect orthogonality (case 2), and capture effect + imperfect orthogonality (case 3). The first case assumes that all spreading factors are perfectly orthogonal and no capture effect. In this case, the gateway can decode all collided packets if the spreading factors are different and none if the spreading factors are identical. This case studies the impact of the data rate allocation only on fairness. The impacts of capture effect and imperfect orthogonality are added in the second and the third cases, respectively.

[1] All details are reported in Table 4.1

When all factors are considered (case 3) the fairness decreases drastically with increasing the network size. Even without capture effect and with perfectly orthogonal spreading factors (case 1), LoRaWAN represents an unfair network because of the unfair collision probabilities associated with the different data rates. Figure 4.1b shows the overall DER of the network. Although the capture effect (case 2) increases slightly the overall DER that comes at the expense of the fairness. However, considering the imperfect orthogonality (case 3), the overall DER is the lowest.

Following these results, a novel mechanism for data rate allocation and TP control algorithm, called *FADR*, is proposed in Section 4.2. *FADR* achieves a fair data delivery for all nodes within a LoRaWAN cell while at the same time being energy efficient. In Section 4.3, detailed results, comparisons and discussions are provided to show and explain how FADR performs under various network configurations. Overall, simulation results show that FADR outperforms the state-of-art in almost all network configurations.

4.1 Related Work

In LoRa/LoRaWAN literature, fairness in DER was not well-investigated with the exception of [RMP17]. In this work, a TP and SF control approach was proposed to achieve fairness in the data delivery. First, the SF ratios are derived to uniform the collision distribution. Secondly, the TP control algorithm is based on the observation that nodes with high path loss and SF8 are the nodes with the highest packet error rate. Therefore, the algorithm assigns a high enough TP to these nodes and allocates SF7 and TP=2dBm, i.e. short airtime and low TP, to all nodes that can corrupt these nodes' packets. Then, the algorithm iterates again over all nodes to allocate enough TP to all remaining nodes. This observation is not always valid as it depends on the node distribution around the gateway, where these nodes may have lower or higher path loss depending on their locations from the gateway. In addition to that, fairness was considered in terms of energy consumption in multi-gateways LoRa network scenarios [GDZ+19, CCG+20].

In the same context, various resource allocation strategies were proposed for LoRa-based networks. In [BRVA16], the SN^5 experiment investigated a strategy of allocating SFs and TPs where nodes choose their transmission parameters locally in order to minimize their transmission times and thus enhancing

the overall lifetime. This strategy represents the standard ADR of LoRaWAN. While in [BR17], a strategy of selecting transmission parameters was proposed to achieve low-power consumption at specific link reliability. Here, a LoRa node probes a link using a transmission parameter combination to determine the link reliability. It then chooses the next probe combination based on whether the new combination achieves lower energy consumption while maintaining at least the same link reliability. The approach terminates when reaching the optimal combination from an energy consumption perspective. Furthermore, two SF allocation algorithms were proposed, namely EXP-SF and EXP-AT, to help LoRaWAN achieve a high overall data delivery rate in [CCC+17]. EXP-SF equally allocates SF to N nodes based on the RSSIs, where the first $N/6$ nodes with the highest RSSIs get SF7 assigned and the following $N/6$ nodes get SF8 and so on. EXP-AT is a more dynamic strategy than EXP-SF, where the SF allocation theoretically equalizes the airtime of nodes, leading to a fair collision distribution. Most of all these works have assumed perfectly orthogonal SFs, although it was shown in [MPJ17, CGT+17] that this is not a valid assumption. Unlike all the aforementioned heuristic approaches, recent works looked at the optimal solution of the resource allocation problem in terms of throughput, energy, and capacity, which is a NP-hard problem [LYF20, PGSDF20, SRHGSGH20, HSMSA20, SQN18]. Also, machine learning and artificial intelligence concepts are employed to solve the resource allocation problem [SGSGH19, TKL+19a, TKL+19b].

While in general data rate management and power control approaches have been well studied for cellular systems and WiFi [Yat95a, SS05], these solutions are not suitable for constrained systems like LoRaWAN. The reason is that cellular approaches require fast feedback and high data rates to work, which are not suitable in LoRaWAN.

4.2 FADR Algorithm

In the following, the proposed mechanism, *FADR*, for fair data delivery allocation and power control to achieve data rate fairness among nodes in a LoRaWAN cell is presented. Firstly, the fair data rate distribution is derived in subsection 4.2.1, which tries to achieve an equal collision probability for all deployed data rates, then the proposed TP control algorithm is presented in subsection 4.2.2, aimed at mitigating the capture and SF non-orthogonality effects.

4.2.1 FADR - Data Rate Allocation

Each transmission parameter combination (SF, BW, with CR (Chapter 2)) leads to a different data rate and thus airtime, which causes different collision probabilities, resulting in unfairness among nodes within a cell. Finding the fair data rate deployment ratios within a cell is therefore crucial.

SF fair distribution ratios were derived in [RMP17] as follows [2]:

$$p_f = \frac{f}{2^f} / \sum_{i=7}^{12} \frac{i}{2^i} \qquad \forall f \in SFs, \tag{4.2}$$

where p_f indicates the fraction of nodes using a specific SF. Equation 4.2 has been derived by equalizing the collision probability of each SF with taking into account the constraint that the sum of all probabilities must be unity $\sum_{f=7}^{12} p_f = 1$.

However, Equation 4.2 does not consider the impact of BW and CR on the collision probability. Assuming that all SFs will be deployed with the same BW and CR may not always be correct as the network operator may consider assigning different BW and CR to the same SF in order to achieve a different data rate, reliability, or sensitivity. Therefore, Equation 4.2 is extended into Equation 4.3 to consider the impact of BW as follows:

$$p_{f,b} = b / \sum_{j \in BWs} j \quad b \in BWs, \tag{4.3}$$

where $p_{f,b}$ indicates the fraction of nodes that use a specific BW from the fraction that use a specific SF. Equation 4.3 is derived with respect to the constraint $\sum_{i \in BWs} p_{f,b} = 1$. In order to also consider the impact of CR, Equation 4.3 is finally extended to Equation 4.4 as follows:

$$p_{f,b,c} = c / \sum_{k \in CRs} k \quad c \in CRs, \tag{4.4}$$

where $p_{f,b,c}$ indicates the fraction of nodes that use a specific CR from the nodes fraction that use a specific SF and BW. Equation 4.4 is also derived with respect to the constraint $\sum_{i \in CRs} p_{f,b,c} = 1$. Combining Equations 4.2, 4.3, and 4.4 gives

[2] The derivation is presented in Appendix A

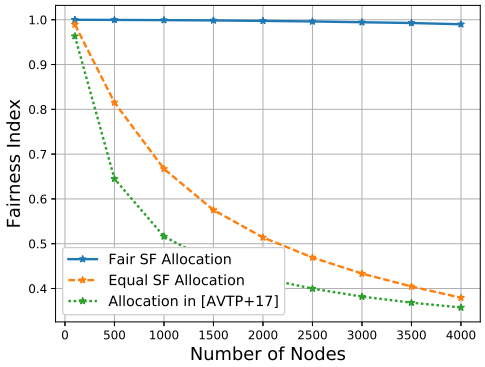
Equation 4.5 that shows the fair ratios of deploying each DR p_d.

$$p_d = \frac{fbc}{2^f} / (\sum_{i=7}^{12} \frac{i}{2^i} \sum_{j=BWs} j \sum_{k=CRs} k) \quad \forall f \in SFs \& b \in BWs \& c \in CRs. \tag{4.5}$$

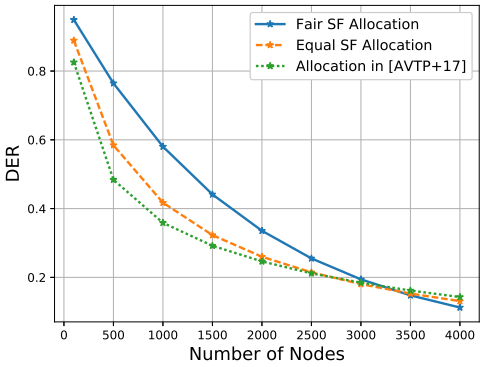
The full derivation of Equations 4.2, 4.3, 4.4, and 4.5 are presented in Appendix A. Equation 4.5 is a generalized form of Equation 4.2, where in case of deploying all SFs with the same BW and CR, the values expressed by Equation 4.5 equal the values derived with Equation 4.2. Hence, the fair ratios of using a potential LoRaWAN data rates without considering CR are: $p_0 = 0.024$, $p_1 = 0.044$, $p_2 = 0.08$, $p_3 = 0.144$, $p_4 = 0.257$, $p_5 = 0.15$, and $p_6 = 0.3$.

Observing Equation 4.5 in allocating the data rates within LoRaWAN cell assures that each node experience the same probability of intra-SF collision. However, this leaves the question as to what is the criteria of allocating data rates over RSSI values within a LoRaWAN cell? BWs and CRs do not affect the orthogonality, while the same does not hold true for SFs, which depend on the received power. Consequently, two allocation strategies can be used for SFs. The first strategy assigns the lowest SF to the highest RSSI values captured from the nodes and so on. This allocation strategy is basically corresponding to the distance from the gateway, where the nearest nodes are assigned the lowest SF. The second allocation strategy is to distribute the SFs along the RSSI values in a cell. In this case, the region concept is proposed. For this a LoRaWAN cell is divided into regions, where each region consists of the same number of nodes. The first region contains the smallest RSSI values and so on. Within each region, all SFs have to be represented and they are assigned using the fair ratios (Equation 4.5). In this case, the fair radio of using SF12 per each region is 2% (the smallest ratio, *see* Equation 4.2). For a discrete representation of the 2%, the recommended smallest region size should be 50 nodes. In this case, SF12 is represented by one node in each region. The two allocation strategies are studied in terms of fairness and energy consumption, and data delivery in section 4.3.

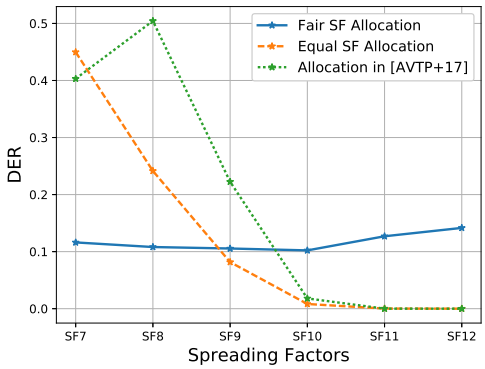
To verify the impact of the fair data rate allocation strategy, this strategy is compared (assuming all nodes are within a single region) to different allocation strategies from the literature. First, equal SF allocation across nodes, which was considered in [CGT+17] and by the EXPLoRa-SF approach in [CCC+17]. Secondly, the proposed allocation strategy in [AVTP+17], where about 28% of nodes use SF12. The fairness is calculated using Jain's fairness index [JCH84]



(a) Fairness Index



(b) DER



(c) DER vs SFs (4000 nodes)

Figure 4.2: Different SF Allocations Study in which $BW = 500Hz$, $CR = 4/5$, and $TP = 14dBm$

(*see* Equation 4.1). The fairness index varies from zero to one, where a higher index indicates a higher fairness. The results are shown in Figure 4.2 for different network sizes. In this study, perfect orthogonal SFs and no capture effect are assumed for a fair compassion. This study [3] provides an insight into the fairness of the data rate allocation strategy regardless of the assigned TPs.

Figure 4.2a shows the fairness index, where the fair allocation is almost one regardless of the number of nodes. However, it dramatically degrades in the other allocation strategies with increasing number of nodes due to increasing collisions. The impact of allocation is clear in Figure 4.2c, which shows DER versus SFs for a cell of 4000 nodes. The DER for nodes using a low SF is higher than for those using a high SF with equal SF allocation and with SF allocation as per [AVTP+17]. DER is almost zero for SF10, SF11 and SF12, which represent half of the nodes for the equal SF allocation. This means, half

[3] All other details are reported in Table 4.1

of nodes cannot deliver any packets due to collisions, whereas the DER for the fair allocation is nearly equal for all SFs and around the random access limit (*see* Figure 4.2b). The overall DER at the fair allocation outperforms the other two allocation strategies up to about 3250 nodes. Beyond this number but this comes at the expense of very poor fairness (Figure 4.2a). The reason behind that is that the overall number of collisions in the case of the fair allocation strategy is exceeding the corresponding numbers in the other two allocation strategies for the sale of equalizing the DER per SF as shown in Figure 4.2c.

4.2.2 FADR - Transmission Power Allocation

Apart from the data rates allocation the other aspect that creates unfairness in LoRaWAN is the near-far problem, which influences the capture effect, especially for non-orthogonal SFs. These characteristics favor near nodes over far nodes, due to the usually higher received power levels of the former. In order to tackle this, balancing the received power levels from all nodes is required. This is typically performed using a TP control algorithm that assigns low TP level to near nodes and high TP to far nodes. Balancing the received power levels eliminates the capture effect and thus achieves a fair data delivery regardless of the distance from the gateway.

The proposed TP control algorithm is shown in Algorithm 1. The algorithm manages a list of nodes (N) with a list of corresponding RSSIs, a list of available TP levels (PowLevels) that can be assigned, and a matrix of CIR of all SF pairs as inputs. To avoid instability due to RSSI fluctuations, the algorithm is run after a certain number of packets have been collected by the network server in order to calculate an average RSSI. RSSI stability has been investigated in [AS14], which showed that the RSSI standard deviation of nodes close to the gateway is less than $3dBm$, however, the deviation increases to $20dBm$ for far nodes. Algorithm 1 does not make assumptions on the initial TP assignment of the collected packets, but recommends that nodes are initiated with the same TP before running the algorithm in order to yield RSSIs with a common reference.

Algorithm 1 allocates a TP to each node. The algorithm starts with sorting the nodes by their RSSI (line 1). Next, it specifies the maximum (MaxRSSI) and the minimum (MinRSSI) values of the measured RSSIs. Also, specifying the minimum value of CIR (MinCIR), which represents the safe operating margin of all SFs (line 2). Subsequently, the algorithm finds the maximum TP (MaxPower) that can reduce the difference between RSSI extremes to below the safe

Algorithm 1: FADR - TP Control Algorithm

input : List of nodes **N**, corresponding **RSSI**, power levels **PowLevels**, matrix of **CIR**

output: $\forall n \in \mathbf{N}, \mathbf{P}[n] \in$ **PowLevels**

Sort **N** by **RSSI**;
$MinRSSI = min($**RSSI**$)$, $MaxRSSI = max($**RSSI**$)$, $MinCIR = min($**CIR**$)$;
PowLevels$.pop(0)$;
for $i \in$ ***PowLevels*** **do**
    $MaxPower = i$;
    if $|MaxRSSI + MinPower - MinRSSI - MaxPower| <= MinCIR$ **then**
        PowLevels $=$ **PowLevels**$[0 :$ **PowLevels**$.index(i)]$;
        $break$;
    else if $i == max($***PowLevels***$)$ **then**
        powLevels$.pop()$;
$MinRSSI = min[MinRSSI + MaxPower, MaxRSSI + MinPower]$;
$MaxRSSI = max[MinRSSI + MaxPower, MaxRSSI + MinPower]$;
for $i \in range(0, len($**N**$), 1)$ **do**
    if $|$***RSSI***$[i] + MinPower| > |MinRSSI|$ **then**
        $MinPowIndex = i - 1$;
        $break$;
    else
        P$[i] = MinPower$;
for $i \in range(len(N) - 1, MinPowIndex, -1)$ **do**
    if $|$***RSSI***$[i] + MaxPower - MinRSSI| > MinCIR$ **then**
        $MaxPowIndex = i - 1$;
        $break$;
    else
        P$[i] = MaxPower$;
$TempIndex = MinPowIndex$;
for $i \in$ ***PowLevels*** **do**
    if $(|$***RSSI***$[TempIndex] + i - MinRSSI| <= MinCIR)$ *and* $(|$***RSSI***$[TempIndex] + i -$ ***RSSI***$[MaxPowIndex] - MaxPower| <= MinCIR)$ **then**
        for $j \in range(TempIndex, MaxPowIndex, 1)$ **do**
            if $|$***RSSI***$[j] + i -$ ***RSSI***$[MaxPowIndex] - MaxPower| > MinCIR$ **then**
                $TempIndex = j - 1$;
                $break$;
            else
                P$[j] = i$;

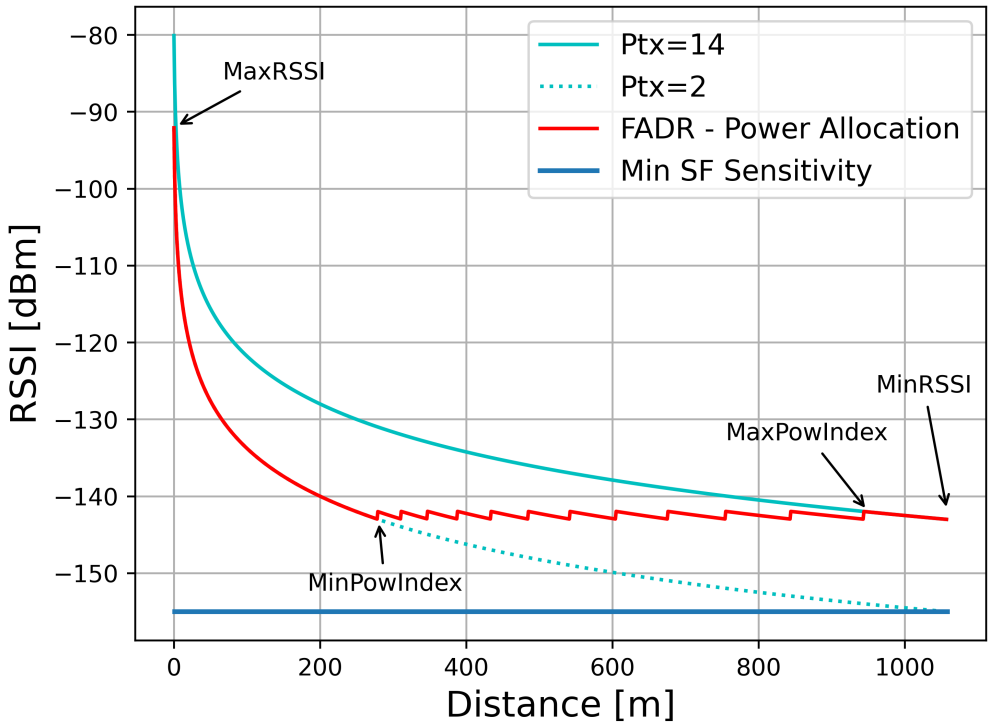


Figure 4.3: Illustration of the FADR Power Allocation

margin (lines 4-10), where the minimum TP (MinPower) is the minimum of PowLevels. In case that MaxPower is less than the maximum of PowLevels, the higher values are removed from the list because they will not be used (line 7). Using lower than maximum power levels, will also reduce energy consumption, thus, extend the nodes' lifetime. Next, the algorithm assigns MinPower to the node with MaxRSSI and MaxPower to the node with MinRSSI, then recalculates MinRSSI and MaxRSSI accordingly (lines 11-12). Subsequently, the algorithm starts allocating the TPs that can be divided into three stages. Firstly, allocating MinPower to high RSSI nodes as long as the new RSSI is not lower than the MinRSSI (lines 13-18). The index of the last node that complies with this approach is saved in MinPowIndex. Secondly, allocating MaxPower to low RSSI nodes as long as the new RSSI plus the safe margin is not higher than MinRSSI (line 19-24). The index of the last node that complies with this approach is saved in MaxPowIndex. Finally, the algorithm assigns to the nodes between MinPowIndex and MaxPowIndex the remaining TPs from low to high as long as the allocation of each TP complies with the rules that the new RSSI plus the safe margin is not lower than the first node using the same TP (lines 26-33).

Figure 4.3 provides a visual explanation of how Algorithm 1 works. In Figure 4.3, MinPower is $2dBm$ and MaxPower is $14dBm$ (the maximum allowed TP in LoRaWAN) since the difference between RSSIs ($\sim 50dBm$ in that example) is higher than the difference between TPs ($12dBm$). Algorithm 1 iterates forward to allocate MinPower until MinPowerIndex, then iterates backward to allocate MaxPower until MaxPowerIndex. Finally, the algorithm iterates in between the MinPowerIndex and the MinPowerIndex to assign the remaining TPs

4.2.3 The complexity and optimality of FADR

In terms of the run time, the algorithm is linearly proportional (i.e., $O(N)$) to the number of nodes per cell since the algorithm iterates over all nodes just once. This is an important property because LoRaWAN is expected to support a very large number of nodes per cell. FADR represents a feasible heuristic approach to the fair resource management problem in LoRaWAN. The heuristic approach has been taken because modelling the optimal problem is very complex. The model depends on the deployment (i.e., device distribution) and the propagation model and without unrealistic assumptions the model cannot be formulated. Due to the expected large solution space (transmission parameter combinations exponent of the number of devices), even with unrealistic assumptions, the run time to find an optimal solution may be to large to be practical.

4.3 Evaluation and Discussion

To evaluate the proposed mechanism, FADR was implemented in LoRaSim [BRVA16]. LoRaSim is an open-source LoRa simulator that takes into account the capture effect only from the same SF, but otherwise assumes perfectly orthogonal SFs. In order to model collisions more comprehensively, LoRaSim was extended to include the non-perfect orthogonality property of SFs based on the work in [CGT+17], which adds a conservative $6dBm$ CIR threshold to all SF pairs. FADR was compared to state-of-art approaches [RMP17] and the SN^5 experiment in [BRVA16] (representing the default LoRaWAN ADR algorithm). The state-of-art approaches are recreated in the used simulation tool and the results are validated with the published results. Multiple experiments are conducted to examine almost all factors that affect the performance. All experiments were run for a real-time of one day and repeated 10 times with

Table 4.1: Simulation Parameters for FADR Study

Parameter	Value	Unit
Nodes [N]	100-4000	
Packet Length [L]	80	byte
Transmission Rate [λ]	60	sec
Max Reception [$MaxRecv$]	8	
Cell Radius [R]	100-3200	m
Channel Number	1	
Channel Frequency [CP]	868	MHz
Simulation Time	86400	sec
Random Seeds	10	

different random seeds.

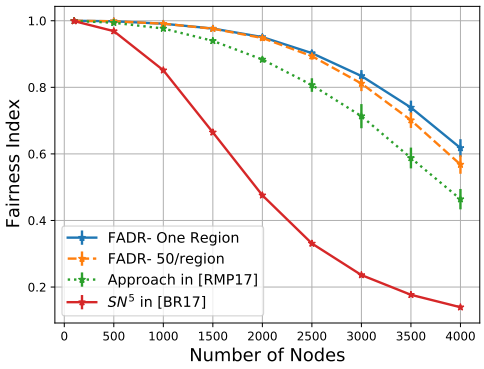
4.3.1 Cell lay-out

In this work, a LoRaWAN cell is considered, which consists of a single gateway located in the cell center and N nodes placed randomly around the gateway. Various cell radii R are investigated. Nodes generate data packets of length L using transmission rate λ. A gateway is able to receive a configurable number of concurrent signals $MaxRecv$, based on its number of LoRa transceivers, on the same carrier frequency CF as long as concurrent transmissions use different SFs and are within the safe margin. For a given combination of SF and BW, packets are only decoded by the gateway if their RSSI is higher than the corresponding sensitivity.

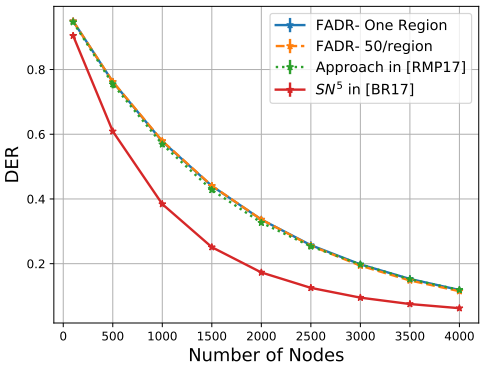
LoRaSim's propagation model was used, which is based on the log-distance propagation model to calculate the RSSI of a node that transmits with TP. The same propagation model is used in [CGT+17] and [MCV17]. Authors of [CGT+17] assume that any node, using any transmission parameter combination, is able to reach the gateway regardless of its distance from the gateway. To achieve the same assumption, the minimum sensitivity of all SF and BW combinations in LoRaSim was lowered to $-155dBm$, so that all nodes can reach the gateway with all combinations. Simulation parameters are shown in table 4.1.

4.3.2 Experimental Evaluation

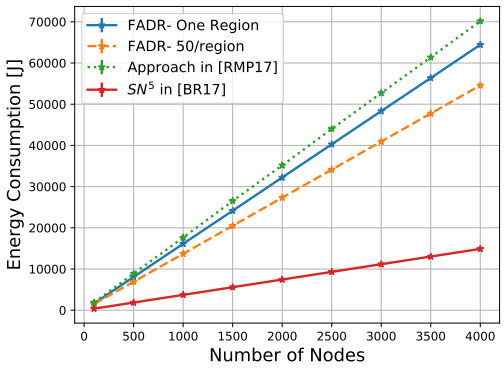
Various experiments were conducted to show the performance evaluation of FADR versus the state-of-the-art. First, the main performance evaluation is



(a) Fairness Index



(b) Overall DER



(c) Overall Energy

Figure 4.4: Main Comparison Results

presented in Section 4.3.2.1. Then, the results are discussed in depth in Section 4.3.2.2. Subsequently, the impact of the cell size is shown in Section 4.3.2.3 and finally the impact of the node distribution is shown in Section 4.3.2.4.

4.3.2.1 Main Comparison

Figure 4.4 shows the overall results of this study. Figure 4.4a shows the fairness index using Equation 4.1, Fig 4.4b shows the overall DER, and Figure 4.4c shows the overall energy consumption. FADR is presented with two region configurations. First, in FADR-One Region, the entire cell is considered a single region. The second approach, where nodes are sorted according to their RSSI and then divided into groups of 50 nodes, was proposed in [ACP18]. The data rate *per region* is allocated based on Equation 4.5 and TP allocation is based on Algorithm 1.

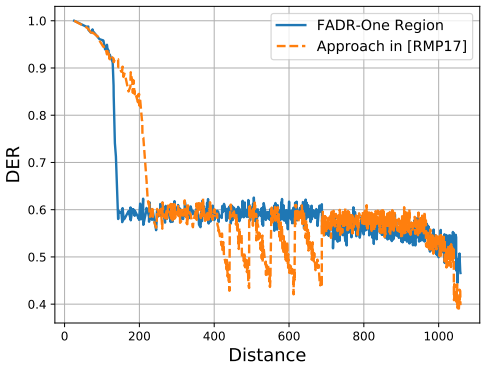
Overall, both FADR region size approaches surpass the other approaches in terms of fairness without sacrificing the overall DER compared to [RMP17] and

with a remarkable improvement compared to SN^5 [BRVA16] (representing the default LoRaWAN ADR algorithm). On the other hand, both FADR region size approaches consume overall less energy than the approach in [RMP17] but more energy than SN^5 [BRVA16], where all nodes choose to transmit using the lowest TP.

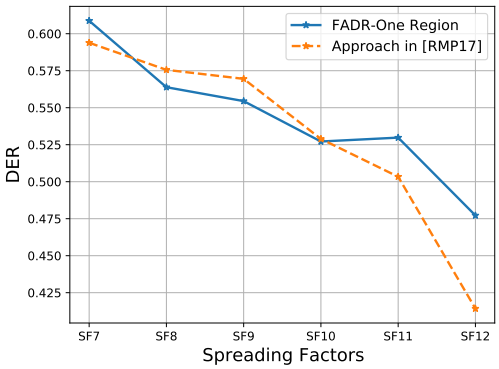
The low fairness and DER performance of SN^5 [BRVA16] is due to the fact that data rate and TP allocation was not studied at the cell level. Rather, nodes choose their transmission parameters locally, which leads to all nodes choosing the same transmission combination that achieves the lowest airtime and using the lowest TP, which in turn achieves the lowest energy consumption, regardless of the cell status. This leads to a degradation of the cell performance by increasing the number of collisions within the cell and leads to aggressive unfairness for far nodes especially when increasing the number of nodes. In the following, the impact of the region size is studied and the performance of FADR versus [RMP17] is analyzed.

The region size has a notable impact on fairness and overall energy consumption, but almost no impact on the overall DER. Decreasing the region size, on one hand, mixes up all SFs in a small variance of RSSI, on the other hand, SFs are distributed everywhere in the cell, not just in contiguous areas as is the case in single region deployment, which allocates low SFs to high RSSIs and high SFs to low RSSIs. Therefore, small regions serve high SFs better in terms of fairness but to the detriment of lower SFs, especially SF7 wherein its imperfect orthogonality impact on the other SFs decreases. While small region deployment increases the impact of the capture effect, especially of SF7, because nodes with the same SF now have high variance in their RSSIs. As overall nodes with SF7 usually represent the majority of nodes in a cell, small region deployment leads to lower fairness index, as shown in Fig 4.4a. Excluding SF7 from the analysis and recalculating the fairness index shows that the small region deployment achieves higher fairness than single region deployment.

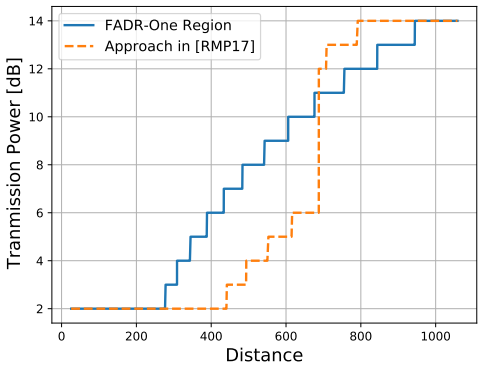
In terms of energy consumption, the FADR TP control algorithm assigns high TPs to low RSSI and vice versa, which leads to single region deployment consuming higher energy than small region deployments. The reason for this is that the nodes with low RSSI, in single region deployment, are allocated with high SFs, i.e. high airtime, and transmit using high TPs, but in small region deployments SFs are distributed over the whole cell, thus, airtimes are distributed over TPs as well.



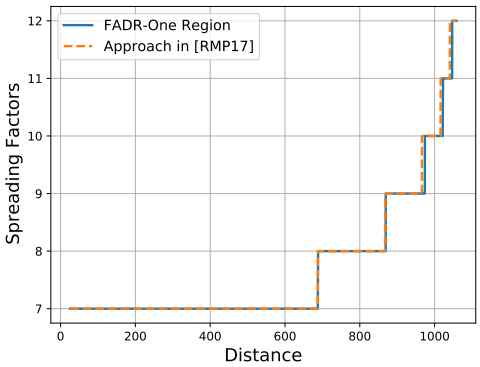
(a) DER vs Distance



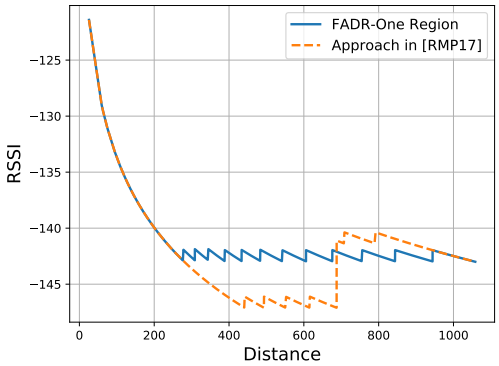
(b) DER vs SFs



(c) Transmission Power



(d) Spreading Factor



(e) RSSI vs Distance

Figure 4.5: Distance Study

4.3.2.2 Distance Study

Figure 4.5 shows DER (Figure 4.5a), transmission powers (Figure 4.5c), SF distribution (Figure 4.5d), and RSSIs (Figure 4.5e) versus distance in addition to DER per SF (Figure 4.5b). These figures provide insights as to why FADR outperforms the approach published in [RMP17]. The results of this study were collected from a cell with 1000 nodes, but performing the same experiment with

a larger number of nodes showed the same behaviour.

FADR's advantage over [RMP17] is shown in Figure 4.5a in which FADR achieves roughly the same DER for a larger proportion of the network compared to [RMP17] while FADR exhibits better fairness. Between 400-700m, Reynders' approach [RMP17] experiences high variation in the DER, corresponding to nodes using SF7 and low RSSIs. These nodes suffer from an aggressive capture effect by other nodes using SF7 and higher RSSIs and at the same time suffer from a capture effect due to the non-orthogonality of SFs from nodes using different SFs and higher RSSIs as shown in Figure 4.5e because they do not get enough TP as shown in Figure 4.5c.

It seems that the TP control algorithm in [RMP17] provides a TP boost to nodes with SF8-12 over nodes with SF7 as shown in Figure 4.5c. This boost yields an advantage to nodes with high SFs over SF7. In this case, as shown in Figure 4.5a), low RSSI values from nodes with SF7 suffer from low DER. In addition to that, this power boost creates a non-orthogonality impact from SF8-9 over higher SFs, leading to low DER for those nodes as shown in Figure 4.5b. Because fewer nodes use SF10-12 than the ones use SF8-9, [RMP17] has slightly higher overall DER, but lower fairness than FADR. This TP boost is also the reason for a higher energy consumption compared to FADR.

On the other hand, the FADR TP control algorithm increases the TP gradually and within the safe margin after reaching the minimum limit of using the minimum TP independently of the SF. This ensures that a large proportion of distances around the gateway have a balanced RSSI within the safe margin. It is worth mentioning that this overall approach is not dissimilar to power control algorithm used in cellular radio systems although these systems usually depend on a continuous TP control mechanisms [Yat95b, GVGZ93]. With FADR, the nodes close to the gateway have an equal impact over the rest of the cell's nodes. This leads to a slight reduction in the overall DER. However, the DER will be more uniform over distance leading to higher fairness as shown in Figure 4.5a and Figure 4.5b.

4.3.2.3 Cell Size Study

The impact of the cell radius on fairness was investigated while keeping the number of nodes constant. Results are shown in Figure 4.6. Increasing the cell radius should provide an increase in the difference of nodes' RSSIs at the gateway. The results shown are collected from a cell with 1000 nodes. However,

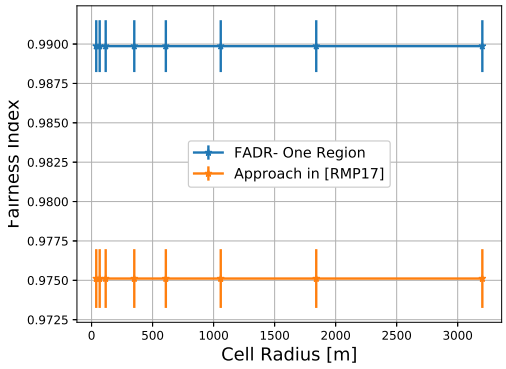


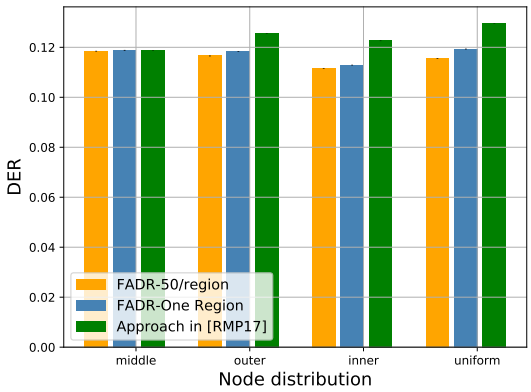
Figure 4.6: Cell Size Study

the behaviour is identical to scenarios with other number of nodes. As shown, the cell radius does not have any impact on the fairness of either algorithms, where the difference is always the same. The reason for this is the slow increase in path loss when moving further away. For example, the difference of the RSSIs experienced at 1Km cell radius is ca. 62dBm and ca. 72dBm for 3Km cell radius. The 10dBm difference between the two cell radii can be handled well within the safe margin of either algorithms.

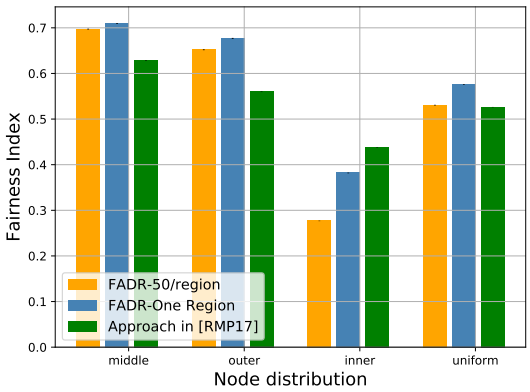
4.3.2.4 Node Distribution Study

The node distribution implemented in LoRaSim was used in all aforementioned studies, which randomly distributes the nodes around the gateway. Changing the node distribution will affect the performance of either algorithms. With respect to [RMP17], it changes the location of the nodes with SF8, which is the reference for their approach. In regard to FADR, it changes the distribution of collisions between nodes. Therefore, to investigate the impact of different node distributions, the cell is divided into three areas (inner, middle and outer), each covering 0.33 of the cell radius. Then the distribution of the nodes was adjusted to allocate 66.6% of nodes to one area and the rest was uniformly distributed in the other two areas. Therefore, the inner distribution, for example, has 66.6% of nodes in the inner disk around the gateway. Figure 4.7 shows the DER (Figure 4.7a), Fairness (Figure 4.7b), and Energy consumption (Figure 4.7c) in the different node distributions. The results shown were collected from a cell with 4000 nodes in each distribution.

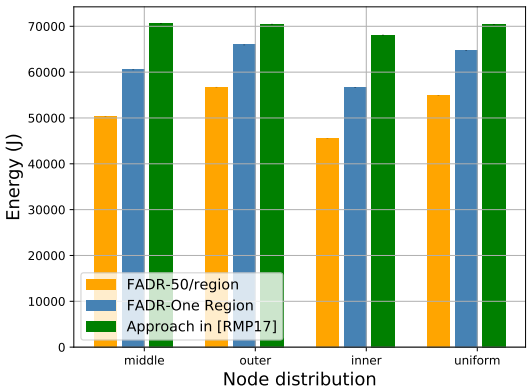
Overall, the results validate the observations made so far that the approach in [RMP17] achieves higher DER, but higher energy consumption and lower fairness than FADR with the exception of the inner distribution, where FADR



(a) Overall DER



(b) Fairness Index



(c) Energy Consumption

Figure 4.7: Node Distribution study

achieves lower fairness. Most of the unfairness in Reynders' approach [RMP17] comes from the impact of the non-orthogonality of low SFs over high SFs in which [RMP17] has higher collisions than FADR. Therefore, FADR from an overall point of view is more suitable for high SFs, i.e. edge nodes, compared to [RMP17]. However, this comes at the expense of DER in the area close to the gateway, where [RMP17] achieves higher DER than FADR.

The inner distribution case is stressful for both approaches because most of the nodes are placed in the region around the gateway with the highest decease rate of the path loss, which affects the remaining 33% of nodes in the rest of the network. The unfairness stems mostly from the impact of non-orthogonality in which [RMP17] has 2.6 times more packets affected by this than FADR. Nevertheless those packets are concentrated in nodes with SF10-12 on the outer region of the network. This is because the power boost in nodes with SF8-9 is now closer to the gateway, creating a big difference in RSSI larger than the CIR threshold for the nodes in SF10-12. Therefore, [RMP17] achieves

slightly higher DER in nodes with SF8-9, but *zero* DER in nodes with SF10-12. Whereas, FADR achieves uniformly distributed DER albeit slightly lower over all those nodes. Due to SF8-9 being used by more nodes than SF10-12, [RMP17] achieves slightly higher fairness than FADR. However, if the nodes with SF7 are not considered, which have much higher DER than all the remaining SFs, FADR achieves 76% fairness, whereas [RMP17] achieves only 64%.

4.3.3 Discussion

4.3.3.1 Scalability of Fairness

From the above studies it should be noted that increasing the number of nodes, which increases the number of collisions, has a negative impact on the cell fairness. As LoRaWAN has a discrete, limited number of TPs (2-14dBm), collisions cannot totally be eliminated in a cell using a TP control mechanism. This leads to collisions being not uniformly distributed over distance, but concentrated in certain areas. It is obvious that the impact of the region of high path loss increase near to the gateway over the rest of the cell. Therefore, increasing the number of collisions magnifies this non-uniformity of collisions, thus, amplifies the unfairness within a cell. Since the transmission rate and the packet length have an impact on the number of collisions as well, these factors affect the fairness as well. In this work, simulation results are based on a generated traffic of 80 byte long packets generated once per minute. However, increasing the transmission rate or packet length with the same number of nodes are found to degrade the fairness as well.

4.3.3.2 Real World Considerations

The effectiveness of FADR TP control in a real world implementation is affected by the variability of the RSSI, which is not totally stable over time [AS14]. Therefore, to avoid RSSI instability, FADR is periodically run after collecting a certain number of packets from each node to average over RSSI samples. The number of packets over which the algorithm should average before running is left for future work. Furthermore, it is known that the RSSI values are highly correlated with the propagation model. However, the propagation model should not aggressively affect FADR's behavior because FADR does not depend on the RSSI values, but the difference between RSSI values, making FADR more relevant for real world implementations than other approaches that depend on the path-loss estimation.

4.4 Conclusions

In this chapter, FADR was proposed as a mechanism to achieve a fair data extraction rate in LoRaWAN cells. FADR achieves this by deploying the fairest data rate ratios that achieve an equal collision probability among the different data rates. In addition to that, FADR controls the transmission power levels of nodes in order to balance the nodes' RSSI values within a safe margin, thus mitigating the capture effect. FADR achieves an almost uniform data extraction rate for all nodes regardless of their positions from the gateway and maintains the nodes' lifetime by not using excessively high transmission power levels. FADR was compared to other relevant state-of-art works for various network configurations, which showed FADR's advantages.

Chapter 5

Reliable and Energy Efficient Data Collections in LoRaWAN

LoRaWAN promises a *deploy and forget* wireless sensing model and the high link budgets of its LoRa modulation [Sem15] make LoRaWAN particularly suited to large-scale data collection applications [HDPMH18] such as environmental monitoring applications. However, LoRaWAN experiences a high collision probability due to its Aloha-based MAC protocol. This results in a high packet loss rate, which affects the network reliability and scalability [BRVA16].

For example, in a common LoRaWAN configuration (spreading factor 12, $125kHz$ bandwidth) the data delivery rate decreases below 50% for gateways serving more than 900 devices [BRVA16]. Even in terms of device lifetime, recent studies [CMVG17] suggest a less than expected performance, e.g., a device lifetime of less than two years can be expected for a transmission interval of ten minutes, or five years for a 100 minute interval. These results are for ideal situations without re-transmissions. The use of acknowledged transmissions to increase the delivery rate surprisingly does not help due to the duty cycle restriction at the gateway, which limits the number of acknowledgments that the gateway can send. This increases the traffic in the network due to the extra packets for re-transmissions, leading to even more collisions, which increases energy consumption and reduces device lifetime without improving the packet delivery ratio [PRKS17]. In summary, collecting data reliably over long periods of time from a medium-large deployment of battery-powered end-devices in LoRaWAN is still a challenge.

In this chapter, a different mechanism to perform data collections using Lo-

RaWAN is investigated. Buffering the data at end-devices and collecting it during scheduled bulk transmissions at convenient points in time slots is proposed instead of transmitting the data as soon as it is generated. The hypothesis behind this approach is that transmissions can be scheduled efficiently in time slots, thus mitigating the collisions and facilitating data compression. Consequently, it can enhance the data delivery ratio and reduced the power consumption. Scheduling transmissions has the potential to eliminate collisions and the use of longer packets reduces the overhead of MAC headers. The proposed approach is called *FREE* - **F**ine-grained scheduling for **R**eliable and **E**nergy **E**fficient data collection. *FREE* is a bulk data collection protocol for LoRaWAN that can support a wide range of delay-tolerant applications such as smart city applications (e.g., vehicle and railway infrastructure monitoring [HOWM14], traffic monitoring [NH16], air quality monitoring [TEG16], smart metering [VS17] etc.), and remote sensing applications (e.g. volcanoes [WALJ+06], glaciers [HTB+08], precision agriculture [Bag05], etc.). Another suitable use case for *FREE* is the data mule [DFDA11], a discontinuous data collection mechanism for large remote geographical areas without wired network connectivity. Here, a LoRaWAN gateway (e.g., carried by a mobile vehicle [ZO18]) is periodically brought into the communication range of the end devices for data collection.

FREE extends the standard simple random access method used by LoRaWAN with a coordinated medium access approach. Specifically, network devices are synchronized and transmissions are scheduled. Although such coordinated access has been investigated before in the general context of wireless networks, it takes a new twist due to LoRaWAN's unique characteristics. LoRaWAN transmissions are restricted by the regulatory duty cycle limitations [ETS12], which is a key constraint for networks operating in unlicensed bands and that do not adopt a *listen-before-talk* policy. In addition to that, LoRaWAN gateways can decode multiple concurrent transmissions (i.e. up to 8) as long as they use different physical channels and/or different spreading factors; although inter-spreading factor interference may appear due to the imperfect orthogonality of LoRa [CGM+18a]. Additionally, downlink transmissions can only take place after uplink transmissions [AVTP+17][1]. These unique characteristics impose challenges on computing the schedule and maintaining synchronization, which have not been studied before in time coordinated networks [DEA06] and, particularly, for bulk data transmission [MLT08].

[1]Here, only LoRaWAN class-A devices are considered

The scheduling algorithm in *FREE* is executed centrally at the gateway based on information collected from end devices, e.g. the amount of buffered data and the actual path loss. The schedule maximizes the throughput by allocating spreading factors, channels, and transmission powers to devices so that concurrent transmissions can be successfully decoded at the gateway. The schedule design utilizes six parallel frames, one per spreading factor, and some of the frames use multiple channels. The channels and transmission powers are allocated such that the impact of imperfect spreading factor orthogonality is minimal. The scheduling algorithm runs in a *greedy online* fashion with the objective to minimize data collection time and energy consumption while obeying duty cycle regulations. Time synchronization, which is crucial for executing the schedule effectively, is maintained before each data collection. *FREE* supports unconfirmable (unacknowledged) and confirmable (acknowledged) uplink data traffic, where the latter one uses a compressed group acknowledgment scheme to work around the duty cycle limitations at the gateway.

5.1 Related Work

The only other work that addresses bulk data transmission in LoRaWAN is [ZAC+19a]*. In [ZAC+19a]*, the authors considered a static network configuration by assuming that the global state information of the network is known before each data collection. In addition to that, the spreading factors are assumed to be fully orthogonal and the authors have not discussed how the synchronization aspect or the acknowledged transmissions can be handled. By exploiting these assumptions, an *offline* allocation was proposed, which is contrary to the online allocation that is proposed in this thesis. Although the offline allocation can achieve somewhat lower energy consumption than *FREE*, it limits the scope of the approach because of its assumptions. In addition to that, it requires a lot of information to be sent by the gateway, which is not realistic due to the duty cycle limitation. In contrast, *FREE* is independent of the network topology and application(s) (i.e. event-based or periodic). Additionally, it realistically handles the scheduling, synchronization and acknowledged transmissions.

Bulk data transmission over wireless channels has been discussed in a variety of contexts. In [LAG+09], an optimized transport layer for bulk data transmission in 802.11 was proposed. Also of interest is [KFD+07] and [GZG12], which provides a bulk data transmission protocol for 802.15.4 and RFID, respectively. In contrast to those works, this chapter presents the bulk data transfer in the

context of LoRaWAN. Supporting bulk data transmissions over LoRaWAN is constrained by various limitations (e.g. duty cycle, multiple non-orthogonal spreading factors, etc.), which is not the case for wireless networks in comparable previous studies.

A part of the contribution is implementing a time-slotted MAC approach on top of the simple Aloha of LoRaWAN. In the literature, similar approaches are considered as they typically offer higher reliability than pure Aloha [BMOG20]. In [LJC18, RWTP+18, GZDJ19], various time-slotted MAC strategies are used to mitigate the collisions in dense LoRa/LoRaWAN networks and thus improve the network scalability. Similar MAC strategies are also considered in [RFF+17, LBPB18, BDF+19, HJY+20] to meet the requirements of soft real-time industrial applications. In [ESRB19, AP19, MK20], time slots are used to enable low-power multi-hop communications in LoRa-based networks to extend the coverage in harsh environments like underground. In addition to that, the Time Slotted Channel Hoping (TSCH) mechanism from the IEEE802.15.4e standard is employed over LoRa to enable LoRa mesh networking in [HOOF20].

A closely related work to the one presented in this thesis is presented in [ZO18]*, where the use of a drone as a gateway in LoRaWAN was suggested for data collection applications. Nevertheless, this work does not enable bulk data transmissions; specifically, end devices transmit their data as it is generated when the gateway is ready. Perfect synchronization between the gateway and end devices is assumed based on a shared global clock. This leads to uncontrolled de-synchronizations and, thus, severe collisions in the long term. In addition to that the duty cycle limitation is not considered in the design and the spreading factors are assumed to be perfectly orthogonal, leading to overly optimistic results. The technique is studied for unconfirmable transmissions only and has the same bottleneck for confirmable traffic type as the standard LoRaWAN.

5.2 System Description

In the following, a private LoRaWAN deployment of one gateway and a number of end devices without interference from other LoRaWAN deployments or other technologies operating in the same frequency band is considered. The devices' applications are sensing or data collection applications, where data is not time critical for a period of time. These applications might generate the data either

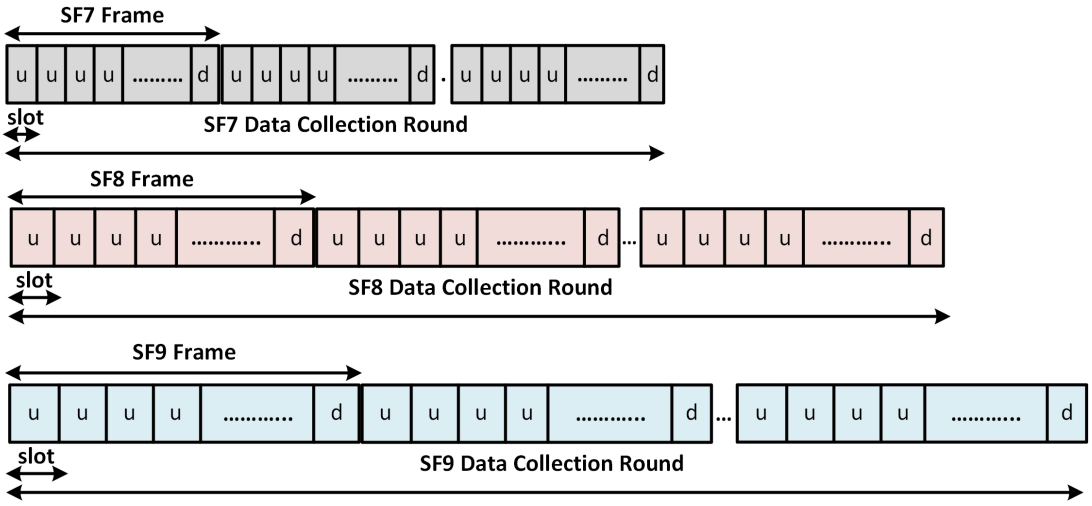


Figure 5.1: *FREE* Frame Structures (u: Uplink and d: Downlink)

periodically or event based or in a mixture of both. In this scenario, the devices can buffer the data in local storage and send it later in bulk. This description matches a wide range of LoRaWAN applications well such as monitoring applications in smart cities [TEG16, VS17] and remote areas [WALJ+06, HTB+08]. Also, it fits use cases where the gateway is only temporarily accessible due to either duty-cycling to save power (e.g. battery-powered) or moving (e.g. data mule [ZO18]).

Once data collection starts, the devices have to transmit their buffered data to the gateway. The amount of this data could be large, depending on the data generation rate and the collection periodicity. With an Aloha-based MAC (without coordination), transmissions would suffer from severe collisions, leading to high energy consumption and prolonged collection time [ZAC+19a]. This is due to the medium becoming almost saturated in this case and, as is well known, asynchronous medium access such as Aloha does not perform well in such a scenario. To alleviate the impact of collisions, transmissions have to be scheduled based on synchronous medium access.

5.2.1 Proposed Solution - *FREE*

FREE is a fine-grained scheduling approach to synchronize transmissions to achieve reliable and energy-efficient data collection. Data collection takes place at periodic/aperiodic intervals known to the gateway and end devices. In order to compute the schedule of each data collection period, the gateway needs to

know the number of end devices that have data to transmit, the amount of their buffered data, and estimate their path loss. The gateway then disseminates the schedule and synchronizes all end devices with the same time reference. Computing and disseminating the schedule and maintaining the synchronization happens in two stages as will be illustrated in Section 5.2.2.

FREE supports three channels of 1% duty cycle as in LoRaWAN plus a channel of 10% duty cycle, allowable in the EU 868 ISM band (refer to chapter 2). The three 1% channels are used for uplink and downlink and the 10% channel is only dedicated for downlink traffic. Before *FREE* has constructed a schedule, the channels are accessed in an asynchronous manner, but once the schedule is calculated and disseminated to end devices, channel access takes place in a synchronous manner.

The transmissions in *FREE* are scheduled in such a way that devices with the same spreading factor transmit sequentially and devices with different spreading factors transmit simultaneously. For this purpose, time is divided into frames, with each frame having a number of *uplink* slots and one *downlink* slot at the end of a frame as depicted in Figure 5.1. There are six *parallel* frames, corresponding to LoRa's six spreading factors ($7 - 12$). Devices per spreading factor are assigned separate slots in the corresponding frame, where the schedule remains the same in the consecutive frames. The term *round* per spreading factor is used to indicate the consecutive frames that are required to collect all data. In addition to the parallel rounds, *FREE* may also support multiple channels per spreading factor. The round, in this case, starts with one slot delay on the second channel, two slots on the third channel and so on. This is because devices cannot transmit simultaneously on multiple channels. Devices (using this particular spreading factor) transmit in each frame on all supported channels, which speeds up the corresponding data collection round. The slot and the frame lengths are equal for the same spreading factor but may not be equal for different spreading factors. The slot length and the number of uplink slots per frame depend on multiple factors, e.g. packet length, number of devices per corresponding spreading factor, etc. All factors that affect *FREE*'s design are examined in Section 5.3.

Before starting the data collection, the network entities (i.e. the gateway and end devices) need to agree on a unified schedule and a time reference in order to synchronize transmissions. These agreements are performed in what is called "the joining and synchronization phase" as depicted in Figure 5.2. The next

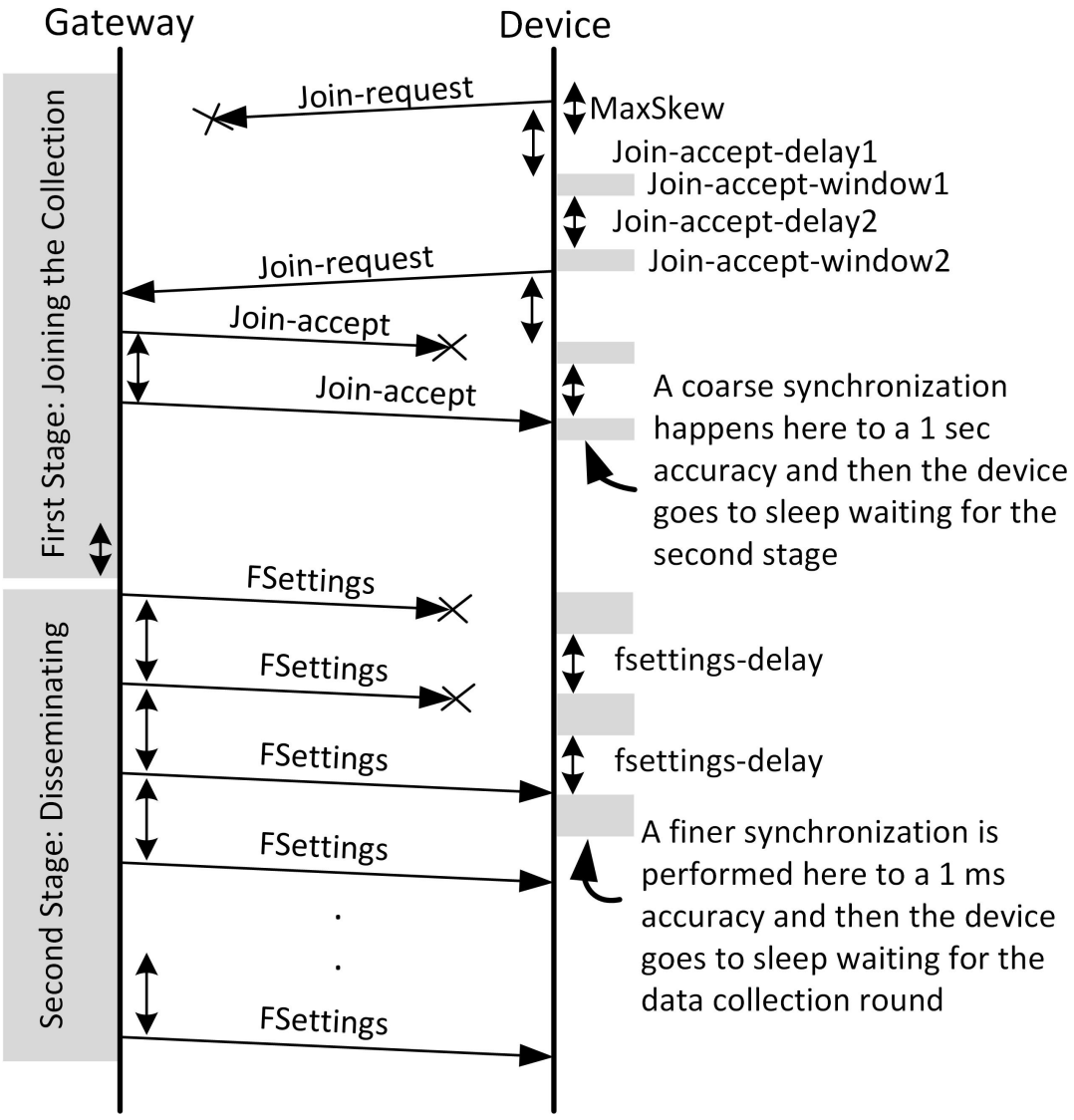


Figure 5.2: The Sequence of Actions in the Joining and Synchronization Phase

subsection describes the joining and synchronization phase in detail.

5.2.2 Joining and Synchronization Phase

This phase consists of two consecutive stages as shown in Figure 5.2. The structure of all messages used in Figure 5.2 is illustrated in Appendix B. The purpose of the first stage is to collect schedule requests from end devices and provide them with their transmission parameters. In addition to that the first time synchronization happens to a $1sec$ accuracy. The second stage uses the gathered information from the first stage to compute and broadcast the schedule for all devices. This stage also increases the time synchronization to less than $1msec$.

The duration and the channel(s) of both stages are assumed to be globally shared among devices in advance.

5.2.2.1 First Stage: Joining the Data Collection

Each device, if it has data to send, submits the intention to transmit in the next data collection round by sending a *join-request* message to the gateway, using the lowest possible spreading factor [ACP18]. In this message, devices reveal the amount of their buffered data *DataSize* and their delay elasticity *DElasticity* in seconds. Once the gateway receives a *join-request* message, it estimates the path loss of the corresponding device and responds with a *join-accept* message. A *join-accept* message guides a particular device to set up its data rate *DataRate*, transmission power *TxPower*, channel(s) *ChMask*, slot number in the assigned frame *SlotFrame*, and the remaining time in seconds to the starting time of the second stage *Secondstage*. Using the *SecondStage* value, devices can synchronize themselves to be less than one second out of synchronization with reference to the gateway. Once a device has received the *join-accept* message, it goes into sleep mode waiting for the second stage.

The *join-requests* are transmitted in an asynchronous fashion using LoRaWAN's default Aloha MAC. In this case, each device keeps sending *join-request* messages until it receives a *join-accept* message from the gateway or exceeds the maximum duration of this stage. These requests are transmitted on the three (1% duty cycle) channels, where a device randomly chooses one of the channels to send its requests. The gateway, if its duty cycle permits, sends the *join-accept* message on the same channel as the *join-request* during the first receive window or on the extra downlink (10% duty cycle) channel during the second receive window.

5.2.2.2 Second Stage: Disseminating Frame Structures

In this stage, the gateway informs devices about their frame structure (i.e. slot length and the frame length) and synchronizes them to a finer time resolution. The gateway broadcasts the *FSettings* message, encoded using the highest spreading factor, during this stage in order to reach all devices. As devices are aware of the starting time of this stage from the previous stage, they can perform a periodic wake up to detect the *FSettings* message. The *FSettings* message contains the packet lengths *PcktSizes*, guard times *Guards* in milliseconds, and the numbers of slots per frames *FrameLens* for all spreading fac-

tors. A device uses the corresponding packet length to divide its buffered data into small chunks equal to this length. Each guard time is calculated based on the worst case de-synchronization that may happen during the corresponding data collection round. Using the corresponding packet size value (*PcktSizes*) and the guard value (*Guards*), each device calculates the slot length using $T_{PcktSizes}+2Guards \ \ [ms]$, where $T_{PcktSizes}$ is the transmission time as defined in [Cor13]. Additionally, the *FSettings* message communicates the remaining time in milliseconds to the starting time of the data collection round *DataCollection* and the remaining time in seconds to the next joining and synchronization phase *NextRound*. Once a device receives one of the *FSettings* messages, it goes into sleep mode waiting for the data collection rounds to start.

5.2.3 Data Collection Rounds

The data collection round, on the same channel, starts at the same time and all devices know about that time from the *DataCollection* field of the *FSettings* message. It should be noted that *DataCollection* is in milliseconds, which is a sufficient time resolution to achieve precise synchronized data collection. In order to transmit its buffered data, each device wakes up for the right slot, waits for one guard time before transmitting a packet and then goes back to sleep mode. Using this method, the transmission takes place within the assigned slot and no overlapping with other slots occurs even in case of de-synchronization among devices. As long as a device has data to send, it will wake up for the same slot in subsequent frames and will repeat the same procedure. It should be noted that the transmissions are performed using the transmission parameters that have been assigned to each device in the *Join-accept* messages.

In case of a confirmable uplink transmission, the device wakes up again before the last slot in the frame to receive the acknowledgment message. If a device could not receive the acknowledgment message due to a transmission error, the device will re-transmit the same packet up to 8 times as specified in the LoRaWAN standard, before dropping it. The gateway acknowledges all received packets during the frame at once using a bitmap message. The bit position in this message corresponds to the slot number in the frame. A bit equalling 1 in the acknowledgment message indicates a correct packet reception in the corresponding slot, otherwise nothing or a corrupt packet was received.

The data collection periodicity (*NextRound*) is subject to the delay elasticity of the running applications (*DElasticity*). Specifically, the data collection has to

Table 5.1: Notations used in Chapter 5

Symbols	Notations
$f \in \mathcal{F}, b \in \mathcal{B}, c \in \mathcal{C}$	Spreading factor, bandwidth, coding rate
$P_{f,\mathrm{BER}}, P_{f,\mathrm{PER}}$	Bit error rate, Packet error rate
L_f, H	Packet length per f, MAC header length
ω_f, SNR_f	Receiver sensitivity, SNR limit per f
T_f	Transmission time of $L_f + H$ packet per f
M_f	Number of assigned channels per f
d	Duty cycle
N_f	Number of assigned devices per f
G_f	Guard time per f
S_f	Number of slots in the frame per f

be performed more often than the lowest delay requirements. For instance, if the network hosts two applications, where the first has a delay elasticity of $12 - hours$ and the second one of $24 - hours$, then the data collection will be performed every $12 - hours$ or less to satisfy the delay requirements of both applications.

5.3 Fine-tuning a *FREE* schedule

The *FREE* approach schedules LoRaWAN transmissions in a manner so as to speed up the overall data collection and minimize the overall energy consumption. However, LoRa transmissions are affected by many factors that are technology-dependent, e.g., duty-cycle, multiple spreading factors, etc. These factors result in interesting trade-offs in the schedule design. In the next subsections, these trade-offs are presented and allocation algorithms for the transmission parameters are proposed. The notations used throughout this paper are summarized in Table 5.1.

5.3.1 Packet Lengths

The packet length (i.e. a chunk of devices' buffered data) is both directly and inversely proportional to the energy consumption. On one hand, longer packets have a higher probability of error, which then requires re-transmissions for successful delivery and, therefore, higher energy consumption. On the other hand, if shorter packets are used, the ratio of payload to MAC header is reduced, so a higher number of transmissions are needed to send a certain amount of data, resulting in higher energy consumption. In this section, this trade-off is ad-

dressed theoretically to compute the packet length that minimizes the overall energy consumption.

First, the Bit Error Rate (BER) of the LoRa modulation is needed for a given spreading factor f as per Equation 5.1 [RMP16].

$$P_{f,\mathrm{BER}} = \mathcal{Q}\left(\frac{\log_{12} f}{\sqrt{(2)}} \frac{E_b}{N_0}\right), \tag{5.1}$$

where E_b/N_0 denotes the energy per bit to noise power spectral density ratio and $\mathcal{Q}(x)$ is the Q-function. E_b/N_0 can be converted into Signal-to-Noise-Ratio (SNR) as follows [Jac12]:

$$E_b/N_0 = SNR_f - 10\log\frac{b}{2^f} - 10\log f - 10\log c + 10\log b, \tag{5.2}$$

where b and c are the bandwidth and coding rate respectively, and SNR_f is the SNR limit per spreading factor f. These limits are given in Table 2.1 chapter 2.

Using Equation 5.1, the packet error rate can be written as:

$$P_{f,\mathrm{PER}} = 1 - (1 - P_{f,\mathrm{BER}})^{8(L_f+H)}. \tag{5.3}$$

Equation 5.3 assumes independently distributed and constant bit error across a packet, considering a packet to be corrupted if one or more of its bits is corrupted. Although this assumption may not always hold in reality, it is a reasonable approach as it yields the worst case packet error rate. In Equation 5.3, L_f is the payload length of a packet using spreading factor f and H is the length of the MAC header in bytes. Given Equation 5.3, the expected number of re-transmissions can be written as [DCABM05]

$$R_f = \sum_{n=1}^{\infty} P_{f,\mathrm{PER}}^n = \frac{P_{f,\mathrm{PER}}}{1 - P_{f,\mathrm{PER}}}. \tag{5.4}$$

Equation 5.4 represents the worst case because the number of er-transmissions is usually constrained and typically equals 8 in LoRaWAN. In this case, the energy consumption of transmitting the total buffered data of size S can be calculated using Equation 5.5.

$$E_f = (1 + R_f)\lceil\frac{S}{L_f}\rceil T_f I V, \tag{5.5}$$

where T_f is the transmission time for sending a packet of length $L_f + H$, and I

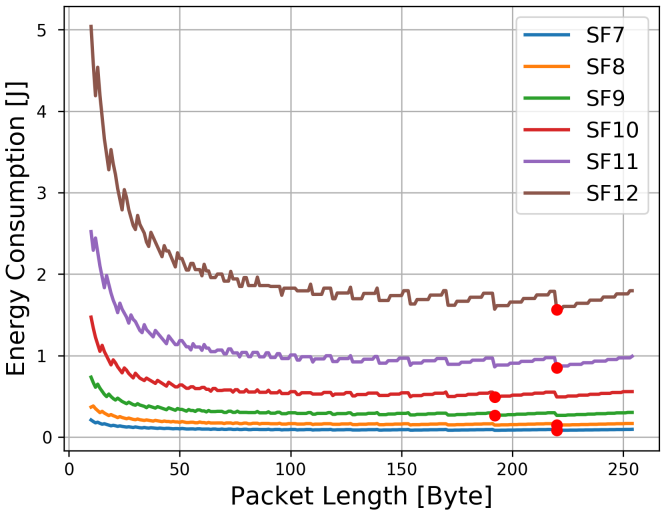


Figure 5.3: Energy Consumption at different Packet Lengths

and V are average current and voltage in the transceiver chip during transmission, respectively.

Figure 5.3 shows an example of the energy consumption in the case of transmitting buffered data of size 1.5 KBytes. All packet lengths are examined, starting from 20 bytes length to 255 bytes, which is the longest packet length permitted in LoRa. The point on each curve represents the packet length that achieves the minimum energy consumption for the corresponding spreading factor. As shown, the packet length for all spreading factors tends to be large to minimize energy. The conclusion from this evaluation is that the impact of packet errors is not as critical as the impact of the MAC header overhead in terms of energy consumption. For this reason, long packets for all spreading factors are better than short packets to reduce the overall number of transmissions and, thus, the impact of MAC headers.

5.3.2 Spreading Factor and Slot Number

The spreading factor allocation affects the overall network performance in terms of energy consumption and data collection time. First, the spreading factor allocation determines how many slots are in each frame, which affects the time of the data collection rounds and, thus, the overall data collection time. Second, the spreading factor determines the transmission time of packets and, thus, determines the energy consumption [ACP18]. Indeed, the energy consumption increases with higher spreading factors. In this section, the spreading

factor allocation is investigated along with the objectives for *FREE* to minimize data collection time and energy consumption.

Generally speaking, the spreading factor allocation is constrained by the specific path loss between a device and its serving gateway as not every spreading factor can be used for any device [AVTP+17]. This is due to spreading factors having different receiver sensitivities, which can be calculated as follows:

$$\omega_f = -174 + 10 \log b + NF + SNR_f. \tag{5.6}$$

The first term denotes the thermal noise per 1 Hz bandwidth and NF denotes the receiver noise figure. The receiver sensitivity will influence the spreading factor allocation as a device must have a higher Received RSSI than the receiver sensitivity associated with the corresponding spreading factor. The standard LoRaWAN spreading factor allocation algorithm meets the receiver sensitivity constraint by usually assigning the lowest possible spreading factor. Although this approach could minimize the energy consumption, it might lead to excessive use of certain spreading factors. This can lead to an unbalanced schedule that is sub-optimal in the data collection time. Therefore, the objectives of minimizing energy consumption and collection time lead to contradictions in terms of the spreading factor allocation. In order to address this, Algorithm 2 is proposed with two different objective functions, which can be triggered with the α flag, e.g. $\alpha \in \{0, 1\}$. When $\alpha = 1$, the algorithm allocates spreading factors so as to minimize the data collection time, whereas, the energy minimization objective is considered when $\alpha = 0$.

Once the gateway receives a *join-request* message from a device, it extracts the amount of data the device wants to send as well as the RSSI of the message. Then, Algorithm 2 runs to determine the spreading factor that the device should use according to the objective of the allocation. Algorithm 2 checks, firstly, the minimum spreading factor that can be assigned based on the minimum sensitivities (line 4). Then, it evaluates the cost function for all *allowed* spreading factors (i.e. equal or higher than the minimum spreading factor) and assigns the spreading factor that achieves the minimum cost. Subsequently, the next available slot in the corresponding frame is allocated to the device (line 8).

Algorithm 2: α-Spreading Factor Allocation and Slots

input : $\alpha \in \{0,1\}$,$\mathcal{F}$, $\omega_f \forall f \in \mathcal{F}$, $T_f \forall f \in \mathcal{F}$, $M_f \forall f \in \mathcal{F}$, H, d, I,V

1 **foreach** $f \in \mathcal{F}$ **do** $N_f = \emptyset$;

2 **while** *Stage 1* **do**

3 **if** *Received a Join-request* **then**

input : $rssi$, $DataSize$

output: Spreading Factor, Slot number

4 $minf = \arg\min_{f \in \mathcal{F}} rssi > \omega_f$;

5 $tempf = \arg\min_{f \in \mathcal{F}} cost$ **if** $f >= minf$ **then**

6 $$cost = \begin{cases} \left\lceil \max(X_f + 1, \lceil 1/d \rceil) \lceil \right. \\ \left. \frac{DataSize}{L_f M_f} \rceil + (M_f - 1) \right] T_f \textbf{ if } \alpha == 1 \\ \lceil \frac{DataSize}{L_f} \rceil T_f I V \textbf{ if } \alpha == 0 \end{cases}$$

7 X_{tempf} += 1;

8 **return** $tempf, X_{tempf}$;

5.3.2.1 Minimum Energy Consumption ($\alpha = 0$)

In the case of minimizing the energy consumption, the following cost function is used to determine the best spreading factor:

$$Cost_f = \lceil \frac{DataSize}{L_f} \rceil T_f I V \quad \text{with } f \in \mathcal{F}, \tag{5.7}$$

This cost function calculates roughly the energy consumption using spreading factor f. The I and V are average current and voltage in the transceiver chip during transmission, respectively.

5.3.2.2 Minimum Collection Time ($\alpha = 1$)

Devices with the same spreading factor are assigned to consecutive slots in a frame, which determines the frame length. In order to minimize the data collection time, balancing the frame lengths over all spreading factors is required to take as much advantage of concurrent transmissions as possible. Therefore, in some network deployments, it is better to switch certain devices to higher spreading factors than what might be optimal from the energy consumption perspective.

However, each frame has a minimum length that cannot be lowered even if there is only one device using the corresponding spreading factor. The mini-

mum length depends on the corresponding packet length and the duty cycle applied. A minimum frame length is required to allow a device to reuse the same slot number in the following frames without violating the duty cycle. Given the set of spreading factors $\mathcal{F} = \{7, ., f, ., 12\}$ and the duty cycle $d \in (0, 1]$, the minimum frame lengths can be written as T_f/d with $f \in \mathcal{F}$. In case of the slot length equaling T_f, the minimum number of slots in a frame should be equal to $1/d$.

In this case, the optimal spreading factor for the device is the one that minimizes the following cost function:

$$Cost_f = \Big[\max(X_f + 1, \lceil 1/d \rceil) \lceil \frac{DataSize}{L_f M_f} \rceil + (M_f - 1) \Big] T_f \quad \text{with } f \in \mathcal{F}, \quad (5.8)$$

where X_f is the number of devices that already have been assigned with spreading factor f and M_f is the number of channels per spreading factor f. The above cost function calculates roughly the data collection time for this device based on the previous knowledge of X_f. $M_f - 1$ represents the required extra slots in case more than one channel is available for spreading factor f because the frame starts with one slot delay in each channel compared to the previous channel.

5.3.2.3 Allocation Optimality

The spreading factor allocation is performed in the first stage. Through this stage, the gateway has only partial information about the network, i.e., received *join-requests* so far. Therefore, the allocation algorithm is performed in a *greedy online* manner based on this partial available knowledge. The online fashion of the allocation is crucial as it makes the proposed approach *independent* of the network topology and application(s) (i.e. event-based or periodic). This is because the allocation algorithm does not constrain the number of devices that can participate in each data collection, nor the amount of their buffered data or their positions from the gateway. This is important, in particular, for the data mule use case, where the gateway might connect to the LoRaWAN deployment from different positions or in a use case, where end devices may move between data collections. Taking into account this independence, *FREE* allocates the *best* possible spreading factor for each device, according to the chosen objective function. In [ZAC+19a]*, an *offline* version of the spreading factor allocation was investigated in which the global state information of the

network is assumed to be known in advance of the data collections. The offline algorithm works by scheduling transmissions in a decreasing order of the spreading factors. Although it achieves shorter data collection time, the improvement over *FREE* is not significant. This is because the *join-requests* from devices with low minimum spreading factor are more likely to be received before those with high minimum spreading factors, which gives the allocation in FREE more flexibility. This results in the schedule in *FREE* to be comparable to the schedule in [ZAC+19a]* even without knowing the global state information of the network.

In [CACP]*, the optimal model for the time-slotted LoRaWAN was developed. The heuristic solution in [ZAC+19a]* showed a very close performance to the optimum solution in terms of data collection time for medium deployment sizes. Moreover, this performance is achieved with a very small computation time which does not exceed a couple of milliseconds. On the contrary, due to the large space of the solution, such as the number of slots, the solver may require multiple hours to find the optimal solution, which is not feasible.

5.3.3 Transmission Power and Channels

FREE exploits concurrent LoRa/LoRaWAN transmissions due to different spreading factors. However, the error rate of concurrently received transmissions is affected by the differing strengths of the received signals. This is because the spreading factors are not fully orthogonal and, thus, cause mutual interference. The orthogonality property of spreading factors has been experimentally studied in [CGM+18a]. Table 2.2 in chapter 2 presents the CIR thresholds among spreading factors. If two signals (using different spreading factors on the same channel) overlap and the difference between their signal strengths is less than the corresponding CIR threshold, the two signals can be successfully received. Otherwise, only the strongest signal can be detected. Therefore, careful CIR is required in order to receive all the concurrent transmissions successfully. As is usual in spread spectrum communications, the CIR can be performed in power (i.e. control transmission power) or in frequency (i.e. assigned channels).

Controlling the transmission power to minimize the cross spreading factor interference would greatly complicate the scheduling. This may lead to different transmission powers per device per transmission, which would require sending frequent power control commands to devices. Complicated transmission power

control is not required in *FREE* as the number of concurrent transmissions on the same channel is controllable. For that reason, varying the transmission power is used less than the channel assignment. There are two constraints that govern the channel assignment. Firstly, the limited number of uplink channels and, secondly, the maximum number of concurrent transmissions that the gateway can handle. Increasing the number of channels per spreading factor expedites the data collection by enabling more concurrent transmissions. However, this may lead to more transmission rejections by the gateway if the number of concurrent transmissions exceeds the maximum capacity of the gateway. In this work, similar to standard LoRaWAN, a maximum of *three* uplink channels and a maximum number of *eight* concurrent transmissions are considered. Eight concurrent transmissions are supported by most LoRaWAN gateways.

Therefore, *FREE* allocates one channel to each spreading factor from 7 to 10 and two channels to spreading factors 11-12. Using this allocation, the gateway receives eight transmissions on all channels at a maximum. The reason behind allocating two channels to spreading factors 11 and 12 is to reduce their data collection rounds by about 50%. As their minimum frame lengths are large, the overall collection time would be negatively affected even if only a small number of devices used these particular spreading factors. However, this leaves the question as to what is the criterion for assigning the spreading factor against the channels?

As pointed out in chapter 4, the majority of cross spreading factor collisions are favoring low spreading factors, particularly 7 over the high spreading factors due to their high CIR threshold (see Table 2.2) compared to the other spreading factors. Therefore, *FREE* assigns a separate channel to devices that use spreading factor 7 (e.g. channel#1). The assignment of channel and transmission power is shown in Table 5.2. Additionally, *FREE* combines the transmissions of spreading factors 9 and 10 with one of the transmissions of 11 and 12 onto another separate channel (e.g. channel#2). The remaining transmissions are assigned the last channel (e.g. channel#3). The reason behind separating spreading factor 8 from 9 and 10 is also the high CIR threshold of 8 over 9 and 10.

The frequency CIR of spreading factor 7 allows its transmissions to use the maximum transmission level (i.e. $14dBm$), which increases the link reliability without adding any cross interference to the other spreading factors. As spreading factors 10-12 have low CIR thresholds over the others, there is no need to

lower their transmission power either as their transmissions are less likely to power-suppress other transmissions. However, spreading factors 8 and 9 have only about $-13dBm$ CIR threshold against spreading factors 10, 11 and 12. Therefore, ***FREE*** lowers their transmission power by $1dBm$ so as to avoid cross spreading factor collisions with transmissions on the same channel. This allocation technique almost eliminates all cross collisions among transmissions as will be shown in section 5.4, which improves the overall packet delivery ratio and energy consumption of the network significantly.

Table 5.2: Transmission Power and Channel Allocation

Channels	**Spreading Factors**	**Transmission Powers [dBm]**
1	7	14
2	9, 10, 11, 12	13, 14, 14, 14
3	8,11,12	13, 14, 14

5.3.4 Packet Lengths, Guard Periods, and Slots per Frame

Algorithm 3: Slot lengths and Number of Slots

input : $P_{f,PER} \forall f \in \mathcal{F}$,$N_f \forall f \in \mathcal{F}$,$MaxSize_f \forall f \in \mathcal{F}$,$M_f \forall f \in \mathcal{F}$,$d$,$H$,$I$,$V$,$Skewrate$

output: $L_f \forall f \in \mathcal{F}$,$G_f \forall f \in \mathcal{F}$,$S_f \forall f \in \mathcal{F}$

for $f \in \mathcal{F}$ **do**
  $L_f = \arg\min_{4<l<255}(1+R_f)\lceil\frac{MaxSize_f}{l-H}\rceil T_l IV$

for $f \in \mathcal{F}$ **do**
  $G_f = \lceil Skewrate\Big[\max(X_f, \lceil 1/d\rceil)\lceil\frac{MaxSize_f}{L_f M_f}\rceil + (M_f - 1)\Big](T_f)\rceil$;
  $minS_f = \lceil(\frac{T_f}{d})/(T_f + 2G_f)\rceil$;
  $S_f = N_f$;
  if $S_f < minS_f$ **then**
    $S_f = minS_f$;

return $L_f \forall f \in \mathcal{F}$,$G_f \forall f \in \mathcal{F}$,$S_f \forall f \in \mathcal{F}$;

During the first stage, transmission parameters (spreading factor, transmission power, channel(s), and slot number) are assigned to each device separately. However, the frame structure and specifically, the number of slots per frame and the actual slot length, are not known by devices at that point because this information requires knowledge of the spreading factor distribution among all devices. At the end of the first stage, the gateway will have complete knowledge of all devices that participate in the coming data collection. Algorithm 3 was

designed to determine the frame structure, which is broadcast periodically in the *FSettings* messages as part of the second stage.

Here, the slot length depends on the packet transmission and guard times, and is equal to the transmission time plus 2 times the guard time. The packet lengths should be selected as explained in Section 5.3.1 to minimize energy consumption. All devices that use the same spreading factor need to have the same slot length to maintain synchronization. Therefore, the slot length is derived from the longest packet length, which, in turn, is computed based on the maximum buffered data size for a particular spreading factor (line 2). Once the packet length is known, it is used to calculate the guard period which should be large enough to accommodate the maximum clock skew accumulation estimate during the previous data collection round (line 5).

With the guard time and the slot length known, Algorithm 3 calculates the minimum number of slots per frame ($minS_f$) that satisfies the duty cycle (line 8). If the actual number of slots per frame (S_f) is less than $minS_f$, it is set to $minS_f$. After receiving the frame structures in a *FSettings* message, each device will know its slot length and the number of slots for its frame.

5.4 Performance Evaluation

To evaluate the operation and performance of *FREE*, a simulation tool, called LoRaFREE, was developed using Simpy and using the log-distance path loss model of LoRaSim [BRVA16]. LoRaFREE is more comprehensive than LoRaSim as it considers a packet error model, the imperfect orthogonality of spreading factors, the fading impact, and the duty cycle limitation at both, the devices and the gateway. In addition to that, LoRaFREE supports bidirectional communication by adding the downlink capability and a re-transmission strategy in case of confirmable uplink transmissions. Furthermore, the energy consumption profile from LoRaSim was extended to consider the consumed energy at reception time. The aforementioned features are required for a proper evaluation of LoRa-based systems making LoRaFREE beneficial to the research community [2].

In the following section, simulation comparisons are presented from the two *FREE* scheduling scenarios (i.e. $\alpha = 1$ and $\alpha = 0$) against two other approaches, namely Legacy LoRaWAN and Delayed LoRaWAN, for both, unconfirmable and confirmable transmissions. In all scenarios, the same application is considered,

[2] `https://github.com/kqorany/FREE`

which is a packet generation application following a Poisson distribution. The difference between the scenarios is in how and when those packets are sent to the gateway. Legacy LoRaWAN uses the standard LoRaWAN MAC, where devices follow the Class-A specification, i.e. Aloha-type MAC, two receive windows after each uplink, etc., and transmit immediately whenever data is generated by the application. In Delayed LoRaWAN, devices buffer the application data and transmit in bulk. The devices, in this scenario, still follow the Class-A specification and do not perform any sort of synchronization before transmission. Only a time offset is used before the first transmission and then the rate of transmission is governed by the duty cycle of the channels. The purpose of the time offset here is to de-synchronize devices in order to reduce systematic collisions. As the data is transmitted in bulk, devices send long packets to reduce the overhead of MAC headers as per Algorithm 3 (lines 1-2). The spreading factor allocation in both the Legacy and the Delayed LoRaWAN scenarios is chosen to minimize the transmission time [BRVA16]. Finally, the devices in the two proposed *FREE* scenarios follow the protocol presented in § 5.2, going through the joining and synchronization phase before starting the actual data transmission.

The scenarios are simulated with a fixed collection period, e.g. one day. In the simulation, all devices use a buffer of data (termed goal buffer) from which packets are extracted and sent to the gateway. The initial size of the goal buffer represents the total amount of application data generated during a collection period. The simulation terminates when all devices have transmitted all data in their goal buffers. For instance, if data is collected once every day and $20B$ of application data are generated every $5mins$, each device would start with $5760B$ to be transmitted. Table 5.3 summarizes all the simulation parameters used in the evaluation. Each simulated study is executed 10 times using different random seeds and the mean across all results is presented along with the standard deviation.

The evaluation results for all scenarios are presented in terms of:

- Network energy and data collection time, where minimizing the two metrics helps the objective of this work.
- Estimated device lifetime, which can be deduced from the network energy consumption.
- Network data delivery ratio, which reflects the throughput of the network

For both *FREE* scenarios, the spreading factor distribution, guard times, and

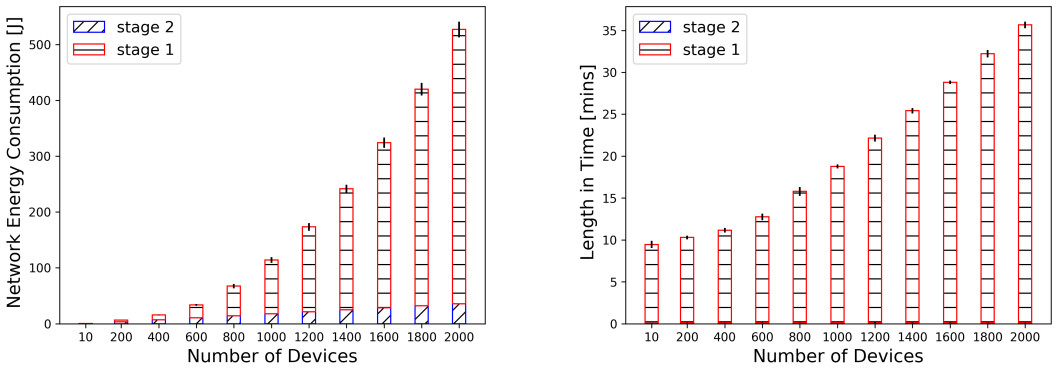
Table 5.3: Simulation Parameters for FREE Study

Parameters	**Value [Unit] + Comment**
Random Seeds	10
Devices	10 - 2000 Randomly scattered
Bandwidth	500 [KHz]
Coding Rate	4/5
Spreading Factor	7-12
Concurrent Receptions	8
Channels	3 Uplink/Downlink with 1% Duty Cycle plus 1 Downlink with 10% Duty Cycle
Retransmissions	8 Times Before Dropping
LoRaWAN MAC Header	7 [Bytes]
FREE MAC Header	8 [Bytes] Extra byte for Data Ordering
LoRaWAN ACK	0 [Bytes] Empty MAC header
FREE ACK	Depends on the number of slots per frame
Path Loss [BRVA16]	$\overline{L_{pl}}(d_0) = 127.41$[dB] $d_0 = 40$[m], $\gamma = 2.08$, $\sigma = 2$
Application	20 [Bytes] every exp(5 [mins])
Collection Periodicity	1 - 48 [Hours]
Transmission Offset	Uniform(0,600000) [ms]
Battery Capacity	1000 [mAh]
Power (Transmission)	132 [mW]
power (Reception)	48 [mW]
Clock Skew Rate	15 [μsec] every [sec]
SNR Limits	Table 2.1
Receiver Sensitivities	Equation 5.6
NF	6 [dB]
Packet Error Model	Equations 5.1 - 5.3

the frame lengths in time are shown. These results help to understand the performance of the two *FREE* scenarios. Additionally, the air time efficiency metric is introduced to indicate the efficiency of the used schedule. This is computed as the ratio between the ideal and the actual data collection time[3]. The metric reflects the sub-optimality introduced by the guard time as well as by the greedy online spreading factor allocation. Finally, the network statistics, i.e. transmissions, collisions, and lost packets, are presented to gain more insight into the results.

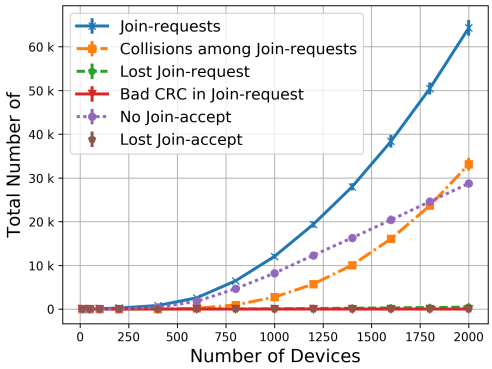
The results from the joining and synchronization phase are presented sepa-

[3]The ideal *FREE* data collection would have each node transmitting only once without any guard periods or idle times, and with frames of the same length. Considering the use of 8 concurrent transmissions, the ideal data collection time length would then be the total time spent transmitting (over all the nodes) divided by 8.



(a) Energy Consumption

(b) Length in Time



(c) Statistics from Stage 1

Figure 5.4: Joining and Synchronization Phase Study

rately in Section 5.4.1. These results provide an insight into the overhead of this phase relative to the network size. Subsequently, performance results for the unconfirmable and confirmable data transmission studies are presented in Section 5.4.2 and Section 5.4.3, respectively. The results of the two transmission types are obtained with a one-day data collection periodicity. Finally, the results obtained by varying the data collection periodicity to serve different data delay elasticity are presented in Section 5.4.4.

5.4.1 Joining and Synchronization Phase Study

Here the energy consumption (see Figure 5.4a) and the time required (see Figure 5.4b) are investigated for devices to go through the joining and synchronization phase. The energy consumption and time required for the second stage increases *linearly* with the network size. This is due to devices performing only receiving activities, which are negatively affected only by losing *Fsettings* mes-

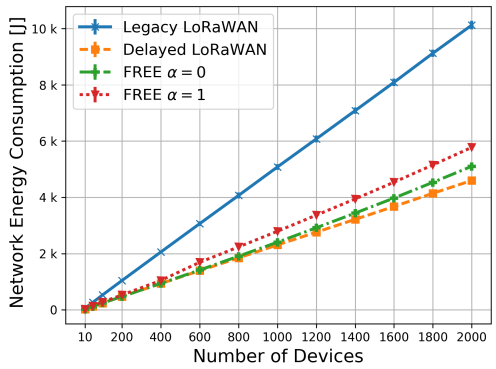
sages due to channel errors. In the first stage, energy consumption increases *exponentially* and the time required for the stage increases *supralinearly* with the the network size. The reason behind this is the scalability issue of the Aloha MAC and the duty cycle limitation of the gateway as shown in Figure 5.4c. For instance, in a network with 2000 devices, the average number of transmitted *join-requests* is 33 per device. That is because roughly 17 of these requests on average collide with other *join-requests* and almost 14 of these requests on average are received by the gateway. However, the gateway cannot send back *join-accept* messages in both receive windows due to its limited duty cycle (see *No join-accept* in Figure 5.4c). In addition to that, a very small percentage of *join-requests* and *join-accepts* are lost due to channel fading.

5.4.2 Unconfirmable Traffic Type Study

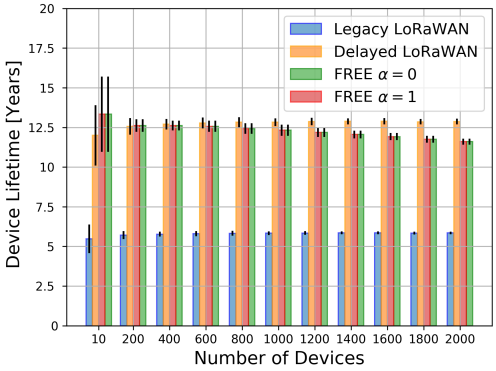
In the following, the unconfirmable traffic is studied, where uplink transmissions are not acknowledged. The results of this study are presented in Figure 5.5 for an application that generates 20 bytes from a Poisson distribution of 5 minutes rate and with a one-day data collection period.

Figure 5.5a shows the network energy consumption for one day and Figure 5.5b shows an estimation of the device's lifetime. The lifetime estimation is calculated assuming a battery capacity of $1000mAh$. Overall, both *FREE* schemes (including the joining and the synchronization phase overhead) consume less energy than Legacy LoRaWAN, but show a higher energy consumption than Delayed LoRaWAN. This is reflected in the device lifetime, where in case of Legacy LoRaWAN, devices survive about 6 years compared to 10 plus years using the other approaches (see Figure 5.5b).

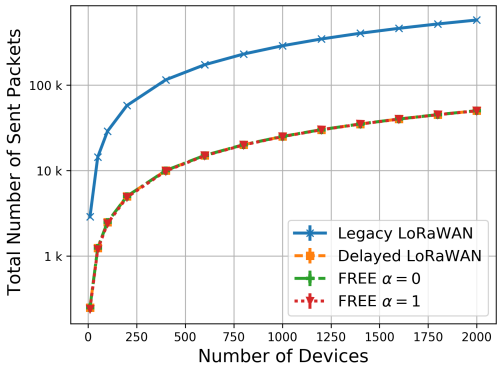
The network size has a *linearly* increasing impact on the network energy consumption and no impact on the devices' lifetime except for the two *FREE* schemes. This is because of the unconfirmable transmissions, where collisions and lost packets do not affect the results. However, the differences among the schemes are in the overhead due to MAC headers and the joining and synchronization phase. In Legacy LoRaWAN, devices do not buffer data but transmit it right away. Therefore, the number of the overall transmitted packets is large compared to the other schemes (see Figure 5.5c) and, thus, the impact of MAC header overhead. For instance, a network with 1000 devices transmits roughly $288K$ packets in case of Legacy LoRaWAN, i.e. $2MB$ of MAC headers, whereas about only $25k$ packets are transmitted in the other systems, i.e., $0.2MB$ of MAC



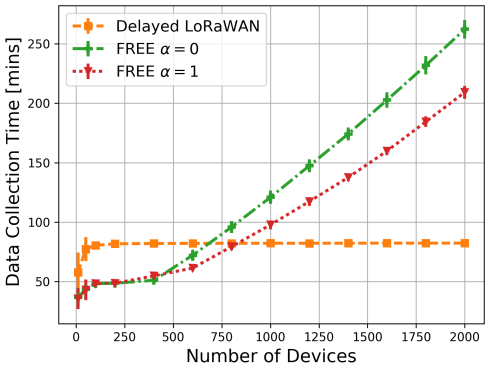
(a) Network Energy Consumption



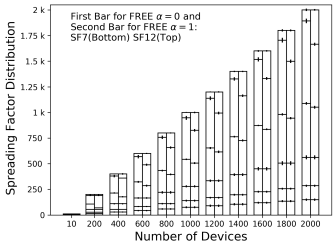
(b) Device Lifetime



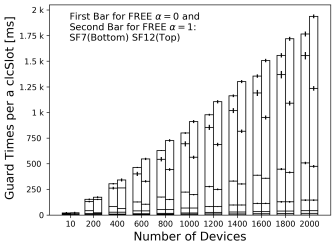
(c) Uplink Transmissions



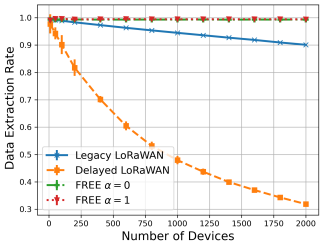
(d) Collection Time



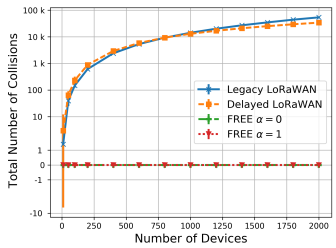
(e) Spreading Factors



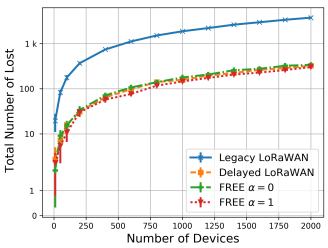
(f) Guard Times per one Slot



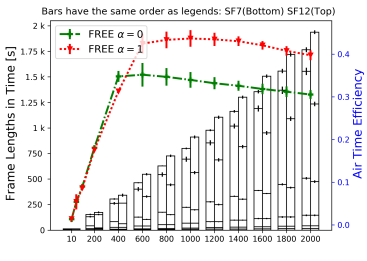
(g) Data Delivery Ratio



(h) Uplink Collisions



(i) Uplink Lost



(j) Frame Lengths

Figure 5.5: Unconfirmable Traffic Type Study

headers. This is the reason why the Legacy LoRaWAN scheme consumes a lot more energy compared to the other schemes.

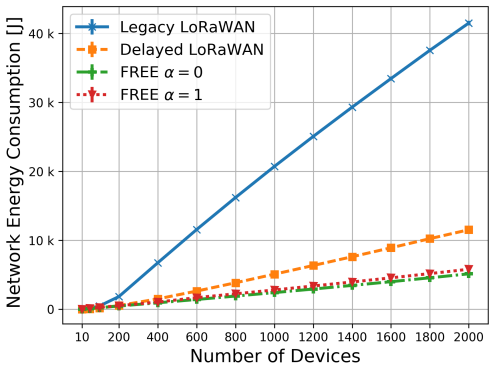
The *FREE* schemes transmit overall the same number of packets as Delayed LoRaWAN. However, because of the overhead of the joining and synchronization phase, Delayed LoRaWAN consumes less energy than the *FREE* schemes. The difference in energy consumption among the two schemes increases with an increase in network size due to the scalability issue of the joining and synchronization phase (Section. 5.4.1). This is also the reason behind the slight degradation in the device's lifetime for the *FREE* schemes.

FREE with $\alpha = 0$ minimizes the energy consumption as shown in Figure 5.5a at the expense of the overall data collection time as shown in Figure 5.5d and vice versa for $\alpha = 1$. This is due to the different spreading factor distributions that each scheme applies in order to optimize the corresponding objective function as presented in Algorithm 2. As shown in Figure 5.5e, when $\alpha = 1$ some devices with low spreading factors have been shifted to higher spreading factors to balance the frame lengths and achieve a lower collection time (see Figure 5.5d). However, this leads to an increased use of higher spreading factors and, thus, a higher energy consumption overall (see Figure 5.5a).

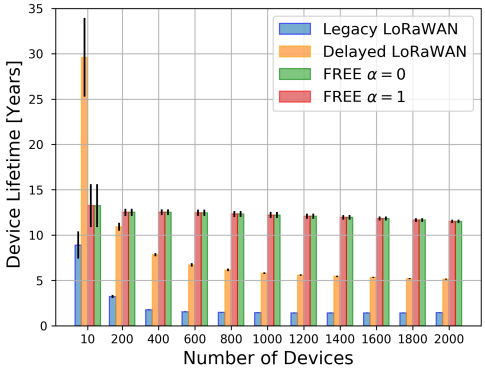
The lowest collection time is obtained with Delayed LoRaWAN[4] (see Figure 5.5d). There are two reasons for these results. The first reason is the time overhead of the joining and the synchronization phase and the second is due to the guard period used in each slot in *FREE*. However, for small network sizes of less than 500 devices, *FREE* achieves less data collection time than Delayed LoRaWAN. This is because devices in Delayed LoRaWAN perform a transmission offset before the first transmission in order to alleviate collisions. Furthermore, for these network sizes, the guard periods are still small compared to the actual times used in transmitting in *FREE* (see Figure 5.5f).

The low energy use and collection time achieved with Delayed LoRaWAN come at the expense of the Data Extraction Rate (DER) (see Figure 5.5g). DER represents the ratio of correctly received data to the initial buffer sizes of all devices over the simulation time. In the case of Delayed LoRaWAN, the DER dramatically degrades by increasing the network size due to collisions (see Figure 5.5h) and channel fading (see Figure 5.5i). Although Legacy LoRaWAN experiences a higher number of collisions and lost packets than Delayed LoRaWAN, it still

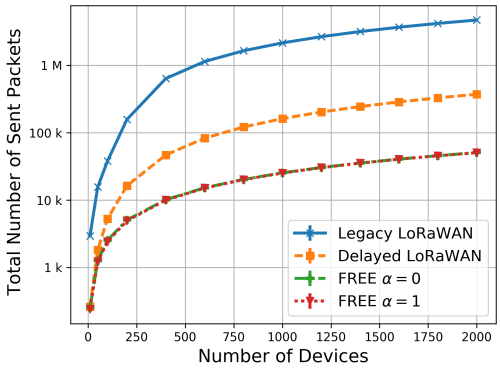
[4]Collection time is not a relevant metric for Legacy LoRaWAN so it is not included in the comparison.



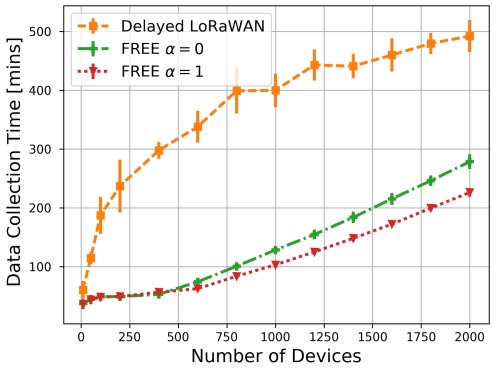
(a) Network Energy Consumption



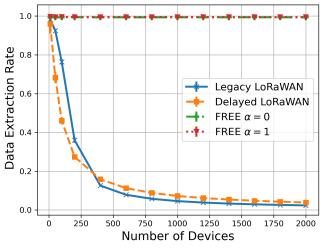
(b) Device Lifetime



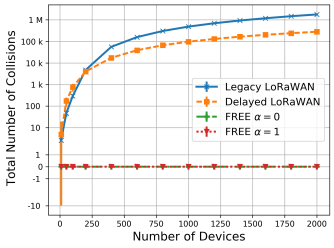
(c) Uplink Transmissions



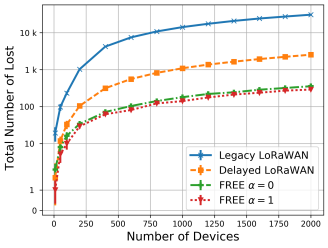
(d) Collection Time



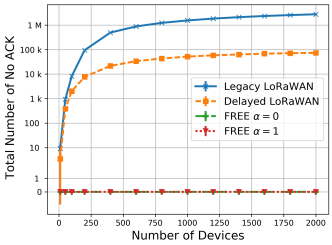
(e) Data Delivery Ratio



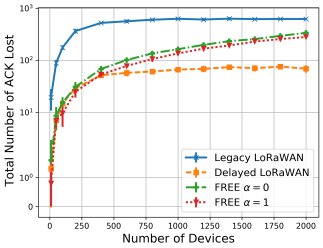
(f) Uplink Collisions



(g) Uplink Lost



(h) No ACK



(i) ACK Lost

Figure 5.6: Confirmable Traffic Type Study

achieves a higher DER. That is because a collision or a loss is more costly in the case of Delayed LoRaWAN due to the large packet lengths. On the contrary, the effectiveness of synchronized communications in preventing most of the collisions in the *FREE* scenarios results in a DER of almost 1 regardless of the network size.

Fig 5.5j shows the frame lengths in seconds for all spreading factors in both *FREE* schemes in addition to their air time efficiency. The frame lengths are directly proportional to the spreading factor distribution (Figure 5.5e), the transmission times, and the guard periods (Figure 5.5f). As shown, the accumulated frame lengths increase for both schemes with increasing network size. In addition to that, the frame lengths associated with high spreading factors are mostly longer than the frame lengths of low spreading factors. That is because high spreading factors require more time to transmit packets of the same length than lower spreading factors. Consequently, high spreading factors require longer guard periods to accommodate clock skew as the time of their data collection rounds is longer than is the case for lower spreading factors (see Figure 5.5f).

The frame lengths are more balanced when $\alpha = 1$ compared to the case of $\alpha = 0$ and, thus, the air time efficiency is higher. It should be noted that the data collection rounds for spreading factors 11 and 12 are running on two channels at the same time. Therefore, roughly half the frame lengths of these spreading factors should be considered for a fair comparison with the other spreading factors. The balanced frame lengths result in a schedule that utilizes the concurrent transmissions as much as possible and, thus, minimizes the overall data collection time. This is the reason that the air time efficiency is higher when $\alpha = 1$ than when $\alpha = 0$. For small network sizes, the air time efficiency of both schemes is low because of the minimum frame length due to the duty cycle. As a result, even if only one device is assigned to a particular spreading factor, the corresponding frame length must be long enough to obey the duty cycle, leading to a lot of non-utilized time in the corresponding frame.

5.4.3 Confirmable Traffic Type Study

Here, confirmable traffic is studied, where acknowledgments are required to provide reception guarantees. The results obtained with this study are presented in Figure 5.6 for the same data generation rate and data collection period as was used in the unconfirmable traffic study. In case an acknowledgment is not received by a device in the expected window(s), the device re-transmits

the same packet up to 8 times before it is dropped. In the case of Legacy LoRaWAN and Delayed LoRaWAN, the acknowledgment message is an empty message with only a MAC header which confirms reception. In *FREE*, the acknowledgment message includes a bitmap of the same length as the number of slots in the frame, as explained in section 5.2.3.

In terms of energy consumption, Legacy and Delayed LoRaWAN schemes are affected badly when the network size increases with a device's lifetime dropping to less than 2 years and to about 5 years, respectively, for a network with 2000 devices (see Figure 5.6a and Figure 5.6b). This is due to the overhead of retransmissions and confirming reception that the network requires in this traffic case. For instance, a network with 1000 devices increases the number of transmissions by 7.4 and 6.4 times, respectively for Legacy LoRaWAN and Delayed LoRaWAN compared to a unconfirmable traffic (see Figure 5.5c and Figure 5.6c). In *FREE*, devices can still survive 10 plus years in both schemes and for all evaluated network sizes. This is because the number of transmissions in this case does not increase much compared to the unconfirmable case due to lack of collisions. A retransmission only takes place due to the loss of an uplink transmission or an acknowledgment because of channel fading. However, these losses happen with low probability.

In the *FREE* schemes, the increase in the overall data collection time is small compared to the unconfirmable traffic (see Figure 5.5d and Figure 5.6d). However, in the case of Delayed LoRaWAN, the difference between the two traffic types is remarkable and gets worse with an increase in network size for the confirmable traffic (see Figure 5.6d) because of the re-transmission overhead. In *FREE*, the spreading factor distribution in this case is identical to the unconfirmable case and this is the reason why the overall data collection time is lower and the overall energy consumption is higher when $\alpha = 1$ compared to the case when $\alpha = 0$.

In contrast, the confirmable traffic does not improve the overall DER in case of Legacy and Delayed LoRaWANs. In fact, the overall number of collisions increases (see Figure 5.6f), which further reduces the overall DER (see Figure 5.6e). However, the collisions are not the main reason for this low DER, but the gateway duty cycle limitation (see. 5.6h). Figure 5.6h presents the number of uplink transmissions that the gateway has received but cannot acknowledge in either of the two receive windows due to the duty cycle limitation. The above observation is in line with the analytical analysis in [PRKS17].

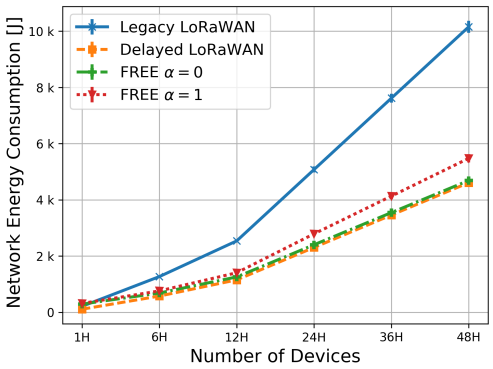
In addition to that, packet loss in the uplink (see Figure 5.6g) and the downlink (see Figure 5.6i) due to channel fading also has a negative impact on the DER, but is not comparable to the impact of the other factors. In Legacy and Delayed LoRaWAN, the DER drops to less than 20% for a network with 300 devices and continues to drop with increasing number of devices, showing the well known scalability issue for LoRaWAN with confirmable traffic. It should be noted that for very small network sizes, Delayed LoRaWAN achieves higher network lifetime than the *FREE* solutions without sacrificing the overall DER (see Figure 5.6b and Figure 5.6e at 10 devices) due to avoiding the overhead of the joining and synchronization phase.

The *FREE* schemes experience almost no collisions which is due to the synchronization and the transmission power control algorithms. In addition to that, the acknowledgments of a complete frame are grouped in one message, which is sent in the last slot of the frame. In this case, the periodicity of this slot is guaranteed to obey the duty cycle of the channel as the frame length is designed to obey it (see Algorithm 3).

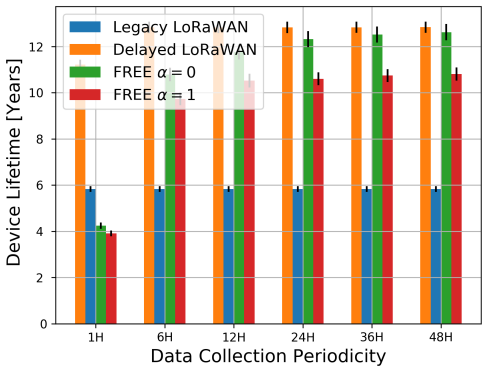
5.4.4 Collection Periodicity and Delay Elasticity Study

Here, the impact of varying the data collection periodicity on the overall energy and collection time is studied, which depends on the delay elasticity of the buffered data. The results are gathered from a network with 1000 devices. The periodicity is varied from $1 - hour$ to $48 - hours$. The results are presented in Figure 5.7 for unconfirmable and confirmable traffic types, respectively.

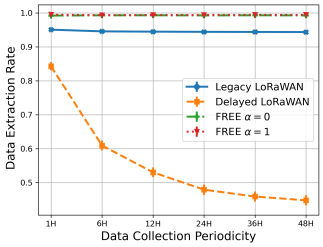
For the unconfirmable traffic type, the *FREE* schemes are more energy efficient than Legacy LoRaWAN until data collections are performed as frequently as once per hour. In this case, Legacy LoRaWAN becomes more energy efficient as it avoids the overhead of the joining and synchronization phase without sacrificing the overall DER (see Figure 5.7c). Although Delayed LoRaWAN is the most energy efficient and fastest scheme for unconfirmable traffic, it comes at the expense of the overall DER. The trend for the confirmable traffic type is consistent; the *FREE* schemes surpass Legacy and Delayed LoRaWANs in terms of device lifetime (see Figure 5.7f) and data collection time (see Figure 5.7h). This happens without sacrificing the overall DER (see Figure 5.7g) for all the presented periodicities. It should be clear that at this network size, Legacy and Delayed LoRaWANs deliver almost no data, which again highlights the scalability problem of Aloha-based systems for the confirmable traffic type.



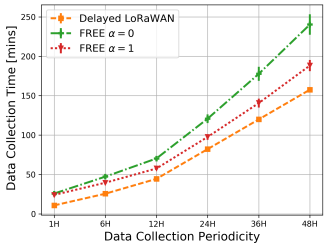
(a) Network Energy Consumption in Unconfirmable Traffic



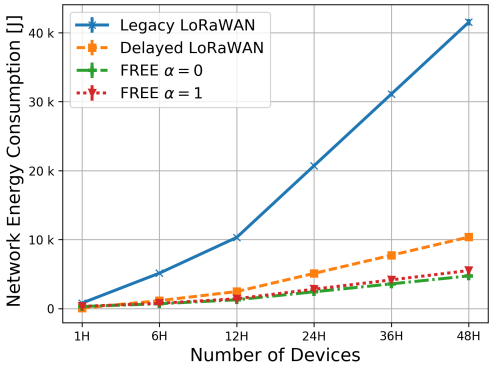
(b) Device Lifetime in Unconfirmable Traffic



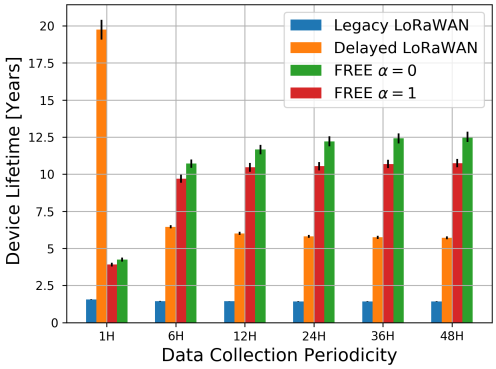
(c) Data Delivery Ratio in Unconfirmable Traffic



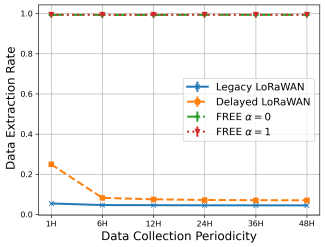
(d) Collection Time in Unconfirmable Traffic



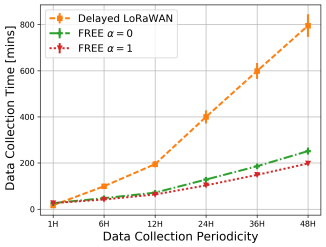
(e) Network Energy Consumption in Confirmable Traffic



(f) Device Lifetime in Confirmable Traffic



(g) Data Delivery Ratio in Confirmable Traffic



(h) Collection Time in Confirmable Traffic

Figure 5.7: Periodicity Study in Unconfirmable and Confirmable Traffic Types

Generally speaking, increasing the collection time periodicity minimizes the impact of joining and synchronization overhead of the *FREE* schemes, which increases the overall device lifetime. However, this also increases the overall collection time as devices have more data to transmit. The opposite holds true in case of collecting the data more frequently. The aforementioned statements are valid for both traffic types and the trend can be seen in (Figure 5.7b and Figure 5.7d) for unconfirmable traffic and in (Figure 5.7e and Figure 5.7h) for confirmable traffic. Therefore, a balance is required to achieve both, a reasonable device lifetime and fast data collection. From this study, a data collection periodicity of $12 - hours$ is found to be a good option. For generalization, data collection should be performed when devices have buffered roughly 2.5 KBytes of data. This amount of data represents a good balance between fast data collection and device lifetime.

5.4.5 Variable Data Sizes

In this chapter, the same application (e.g., periodic update) is assumed to be run on all devices. Resulting in the same amount of data to be buffered on all LoRaWAN devices. This is not typically the case in real-word deployments, where multiple applications may co-exist. The impact of different data sizes on the performance is studied in [ZAC+19b]*, which represents an event-based application or multiple applications on the same deployment. The data size of each device is randomly allocated from $[1000 - X/2, 1000 + X/2]$ bytes, where $X = [100..500]$. The energy consumption and the packet delivery ratio are found to be almost the same as the respective results in the fixed data sizes study. However, in terms of the time, a different behaviour is found compared to the fixed data sizes study. Because there is variation in the amount of data available per device, some devices finish transmitting sooner than others. As a result, devices that complete sooner leave empty, unused slots. In this case, the data collection only finishes when the device with the most available data finishes. This becomes inefficient when the frame is longer than the duty cycle, in which case those slots could be used instead by other devices' transmissions. To tackle this issue, a new scheduling approach is proposed in [ZAC+19b]*, called *Global*, which allows transmissions to be scheduled in different frames. In this case, other devices, with remaining data, can reuse these empty slots. As a result, *Global* can finish the whole data collection sooner.

5.5 Conclusions

In this chapter, the bulk data transmission approach over LoRaWAN is addressed. This approach is ideal for various data collection applications in LoRaWAN, however, it is hitherto unreported in the literature. For that, *FREE* is proposed as a fine-grained scheduling scheme for reliable and energy-efficient data collections. In order to compute *FREE*'s transmission schedule, an overhead phase is required to manage the allocations and synchronize the network. The impact of this overhead is marginal and is minimized when the collection periodicity is large in time. *FREE* schedules concurrent transmissions by using different spreading factors without collisions and by grouping acknowledgments, which are both bottlenecks of the standard LoRaWAN. *FREE* was simulated and compared to two other approaches for unconfirmable and confirmable transmissions. The simulation results showed that *FREE* scales well, achieved almost 100% data delivery and over 10 years battery lifetime independent of the transmission type and network size. This is compared to poor scalability, low data delivery and device lifetime of fewer than 2 years in the case of confirmable traffic type for standard LoRaWAN configurations.

Chapter 6

Firmware Updates over LoRaWAN

Firmware Updates Over The Air (FUOTA) defines the process of updating a device's firmware over a wireless medium. This is a crucial feature for many embedded systems, including any large-scale wireless sensor network installations as it allows to deploy security, optimization, and/or new-functionality patches without much human intervention in order to protect devices, extend their lifetime and/or enhance their performance [Jon18]. In LoRaWAN [LoR17b], FUOTA is an even more critical requirement because of the long device lifetime that LoRaWAN promises, e.g., 10 years [HDPMH18]. This long lifetime stands in contradiction to the fast-changing modern software life-cycle and the LoRaWAN standard, which is subject to continued development. Therefore, FUOTA represents a way to keep LoRaWAN devices up-to-date with the standard throughout their lifetime, which important for reliable, safe and secure long-term operation.

The nature of FUOTA requires downloading a big block of data (e.g. a few hundreds of kilobytes) to the devices. This is a challenging task in LoRaWAN because of the limitations of the communications technology itself [AVTP+17]. LoRaWAN is a low data-rate technology, offering at most a few 10s of kbits/s. LoRaWAN also operates in the unlicensed sub-GHz band, where transmissions have to obey a duty cycle restriction, for instance, 1% in Europe. In addition to that, LoRaWAN is designed for applications with predominantly uplink transmissions. For instance, a downlink transmission, in case of a class-A device, is only available after an uplink transmission. All these limitations challenge efficient FUOTA over LoRaWAN.

In order to better understand the impact of LoRaWAN limitations on FUOTA, let

us consider how to transmit a firmware image of 50 kBytes using DR2 (SF10/ 125 kHz). Even if the maximum packet size (i.e. 51 bytes for DR2) is used, about 1004 downlink packets are required to transmit the whole firmware. For this, a similar number of uplink transmissions is required to solicit the downlink transmissions. Even with a perfect wireless channel with no losses, the firmware update would take ca. 17 hours to upgrade only one device because of the 1% duty cycle limitation. Consequently, updating a medium-size deployment could take up to a few weeks, which is not practical.

The above example points to a number of features that are required for LoRaWAN to support efficient FUOTA:

- A mechanism to send downlink transmissions without the need for uplink transmissions to be sent first. This optimizes the devices' duty cycle and, thus, their power consumption.
- Multicast support in order to optimize the gateways' duty cycle by sending downlink transmissions to multiple devices simultaneously.
- A mechanism to download a big data block and recover packet losses without congesting the medium with transmissions to request the missing packets.

For this, the LoRa Alliance, the industry body behind the LoRaWAN standard, created the FUOTA working group to define the baseline needs to enable efficient FUOTA over LoRaWAN [FUO19]. This has resulted in new specifications to cover multicast, fragmentation and time synchronization topics, which are essential features for efficient FUOTA.

In this chapter, these new LoRaWAN specifications are reviewed and it is examined how the new features can enable fast and efficient firmware update. Additionally, the proposed FUOTA process is analyzed in order to quantify the impact of the different parameters and show the trade-offs among them. In order to analyse the process, a new simulation tool, FUOTASim, was developed to study the FUOTA process in LoRaWAN. FUOTASim can support LoRaWAN operators to determine the best parameters when performing FUOTA.

6.1 Related Work

The need for FUOTA has been recognized since the early days of Wireless Sensor Networks (WSNs) whereby the deployment scale and the often remote and

inaccessible locations of devices were the main drivers behind this need [BS06]. Consequently, a lot of research has been carried out, covering different aspects of the firmware update, including protocols for disseminating the update, reducing the size of the update, and executing the update on the devices [vS13].

The protocols developed for disseminating the firmware update in WSNs are based mainly on the underlying network architecture and protocol stack. The Trickle algorithm is used to send software updates in lossy shared links [LCH+11]. In [HC04], the firmware update is disseminated through multiple paths in multi-hop WSNs. While in [ZHR+18], the update is targeting network stack modules to reconfigure the network on the fly. The Message Queuing Telemetry Transport (MQTT) protocol is used to disseminate the firmware updates to a fleet of WiFi devices in [FRP17]. For WSNs that enable end-to-end IP connectivity (e.g. 6LoWPAN [SYY+13]), the update can be disseminated using Constrained Application Protocol (CoAP) over UDP [SHB14].

This work looks at LoRaWAN in contrast to those other network stacks. LoRaWAN does not support end-to-end IP connectivity and its downlink capability is limited. In addition to that, LoRaWAN's physical layer supports multiple data rates and the transmissions are restricted with the duty cycle of the sub-1GHz ISM band. These limitations hinder the adoption of legacy protocols for disseminating firmware updates over LoRaWAN. It should be noted that the protocols for FUOTA suggested in this work are network agnostic and, thus, they can be directly adopted in similar networks to LoRaWAN such as Sigfox [1].

6.2 Key Requirements of FUOTA

The FUOTA working group defined the baseline needs to perform FUOTA over LoRaWAN. These needs have been described in new specifications that are highlighted here. The objective of the new specifications (mutlicast, fragmentation, and clock synchronization) is to standardize this essential process, leading to an interoperable FUOTA solution. It is worth mentioning that the new specifications are not part of the LoRaWAN MAC but run at the application layer.

Fig. 6.1 shows the recommended FUOTA architecture [FUO19], where interfaces with solid lines are described in the LoRa Alliance specifications, otherwise, they are out of the LoRa specifications scope. The firmware update server (right-hand side) together with the firmware update management initiate and

[1] `https://www.sigfox.com/en`

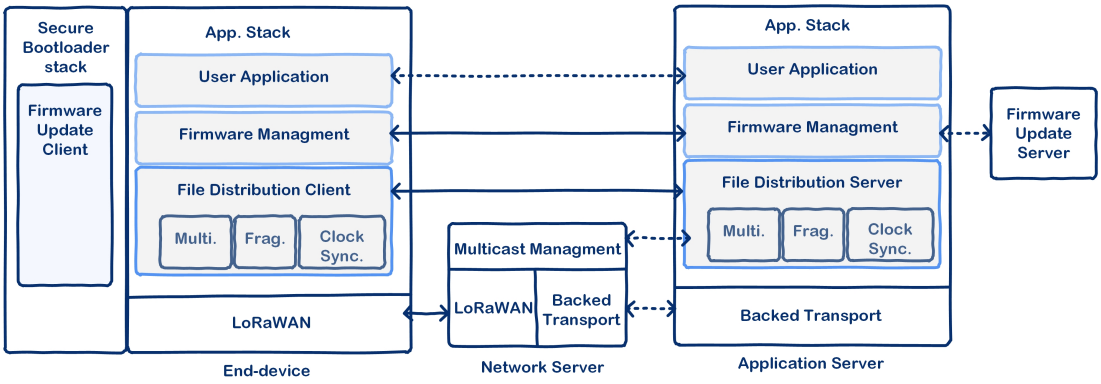


Figure 6.1: FUOTA Architecture

run the FUOTA process for a list of devices. These two components are the brain of the FUOTA process that generates the firmware image and controls the process and the file distribution server. The file distribution server is an application that is tasked to deliver the firmware image to a group of devices, using the underlying features, i.e., multicast, fragmentation, and clock synchronization. These features are highlighted later in this subsection. The architecture also shows the network server in the middle, which handles all communication between LoRaWAN devices and application servers. In addition to that, the network server manages the multicast, including creation, deletion and/or editing of multicast groups and assures delivery of multicast downlink transmissions to all devices in a group. This is performed by calculating the minimum set of gateways that have to send the same multicast transmission to cover all devices in a group.

On the devices side, the counterpart components of the firmware management and the file distribution server are present, along with an additional stack for a secure bootloader, which is essential for any device performing firmware updates. The bootloader is responsible for checking the availability and the integrity of a new firmware image and overwriting the old firmware with the new one. This part of the process is out-of-scope of this paper. However, the work-in-progress of the Software Update for IoT (SUIT) working group [2] is worth mentioning. SUIT is chartered by the Internet Engineering Task Force (IETF) to standardize a manifest that provides meta-data about firmware images (such as firmware identifiers, the hardware needed to run a firmware, and dependencies on other firmware), as well as cryptographic information for protecting a firmware image in an end-to-end fashion. The new solutions are mainly target-

[2] https://datatracker.ietf.org/wg/suit/about/

ing constrained IoT devices similar to LoRaWAN devices [ZSA+19].

6.2.1 Multicast

The objective of the multicast design is to let a group of class A devices receive the same downlink transmission at the same time. This requires that the group of devices is in a receive mode at the same time and share the same security keys to be able to decrypt the same downlink transmission. For this, the multicast specification defines a command to program a receive distribution window of class C or class B into a group of class A devices. Additionally, the specification defines commands to instruct the group of devices to switch to class C or B temporarily at the beginning of the receive window. Then, switch back to class A at the end of the receive window. All commands of this specification are sent to each device individually using unicast messages and 200 as a port number (FPort in 2.8).

Multicast command *McGroupSetupReq* is sent to a device to set up a multicast group. The multicast security key is sent in this command to be used by all devices in the group to derive the multicast security application and network keys. These keys are used to encrypt the multicast messages, so by sharing the same keys each device in the group can decrypt the same multicast messages. *McGroupSetupAns* is sent back by all devices to acknowledge the multicast setup. Subsequently, command *McClassCSessionReq* is sent to a device to indicate that the group that has been set up earlier is a class C group. In addition to that, the command defines the session time, the session time out, the data rate and the channel. The session time indicates when the device has to start the class C receive window. The time is expressed as the time in seconds since 00:00:00, Sunday 6$^{\text{th}}$ of January 1980 (start of the GPS epoch) modulo 2^{32}. The session time out indicates the maximum length in seconds for a device to stay in class C before reverting to class A. The data rate and the channel are the resources that will be used to send the multicast transmissions. For a class B multicast group, command *McClassBSessionReq* is sent, which is similar to the *McClassCSessionReq* command, except in this command, the slot ping periodicity of class B is defined, which indicates how many ping slots are assigned and their periodicity for multicast transmissions. For acknowledgments, *McClassCSessionAns* and *McClassBSessionAns* are sent back by a device to acknowledge *McClassCSessionReq* and *McClassBSessionReq*, respectively.

6.2.2 Clock Synchronisation

LoRaWAN devices usually do not have access to accurate clocks. Consequently, due to clock drifts, their time keeping is not reliable enough to perform *McClassCSessionReq* and *McClassBSessionReq* commands. Therefore, the clock synchronization specification defines a way for the devices to correct their clock skews. The basic idea is that the network has access to an accurate GPS clock that can be used to correct the devices' clocks. All commands of this specification are sent as application messages and 202 port number is used to distinguish the application. Command *AppTimeReq* is sent by a device to ask for a clock correction. The command includes the device time, which indicates the current device clock. The time is again expressed as the time in seconds since 00:00:00, Sunday 6$^{\text{th}}$ of January 1980 (start of the GPS epoch) modulo 2^{32}. Next, the device gets *AppTimeAns* back, including the time correction that stipulates the time delta in secs. The expected accuracy of this approach is around one second, which is enough to run the multicast commands efficiently.

6.2.3 Fragmentation

A firmware image is usually quite big (i.e. a few hundreds kBytes), which cannot fit into one downlink packet but needs quite a number of packets. LoRaWAN links are lossy and thus packet losses are inevitable. Consequently, there is no efficient way to know which packets were lost at which devices during multicast transmissions. Therefore, a mechanism to handle big data blocks and to recover packet losses in a scalable manner is required. For this, the fragmentation specification supports all necessary commands to transport a large data block to one device or to a group of devices reliably if multicast class C or class B is used [FUO18]. All commands of this specification are sent as application messages and port number 201 is used to distinguish this application.

Command *FragSessionSetupReq* is sent to a device to define a fragmentation session. The command specifies which multicast groups are allowed as input for this fragmentation session. In addition to the number of fragments, the fragment size, the fragmentation algorithm (recovery algorithm), and the padding size are specified. The padding size is used as the firmware image may not be an integer multiple of the fragmentation size. A device sends *FragSessionSetupAns* back to acknowledge setting up the fragmentation session.

At the time of writing of this paper, only one fragmentation algorithm was de-

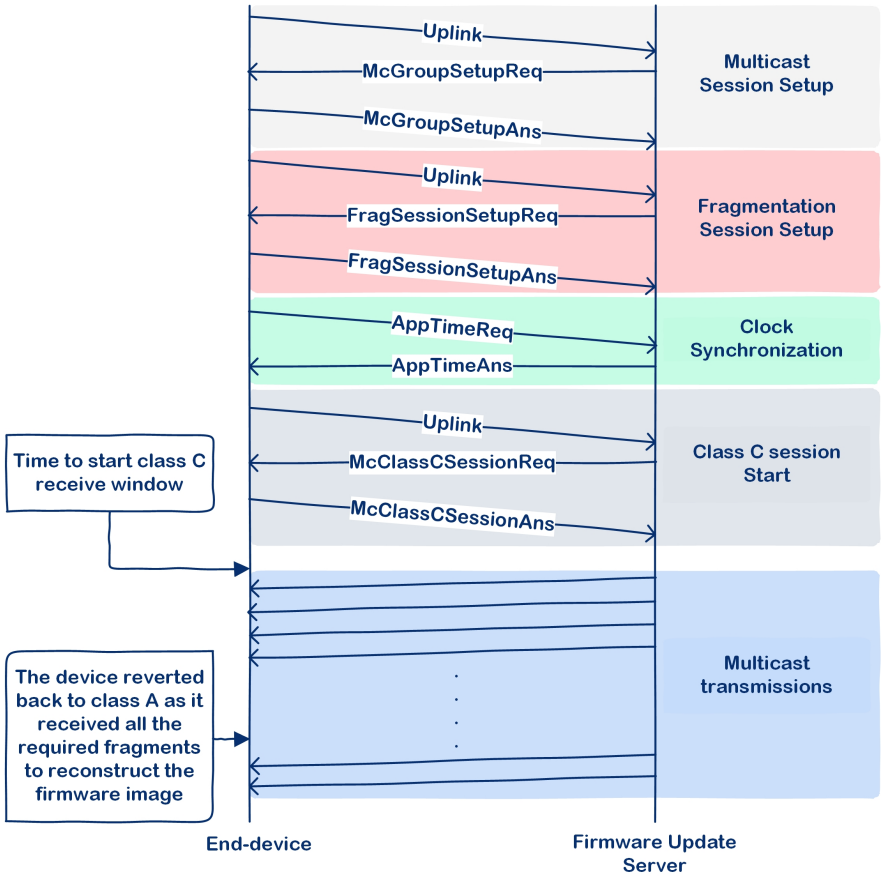


Figure 6.2: FUOTA session using class C multicast

fined. The algorithm proposes adding a simple forward error correction code to the original firmware image before sending it. This allows devices to *autonomously* recover a certain ratio (based on the code used) of the lost transmissions without requesting re-transmission of lost fragments. This is done by, first, chunking the original firmware image to fragments equal in size and then adding redundancy fragments, which are XORed to some of the original fragments. Devices can use redundant fragments to reconstruct their missing fragments. In this case, 5% redundancy added to the original firmware image allows devices to loose roughly 5% of the incoming transmissions and still be able to reconstruct the original firmware [FUO18].

6.3 FUOTA Process

The FUOTA process is initiated at the firmware update server, which generates the new firmware fragments along with the redundant fragments. Also, together with the firmware management, the firmware update server assures

that the required sessions (multicast, fragmentation and etc.) are already established at the intended devices before sending those fragments. The firmware update server takes system decisions which affect the efficiency of the process. These decisions include the topology of the multicast group(s), the class of the multicast (i.e. C or B), the data rate and the channel to be used to transmit the multicast fragments. For example, the firmware update server may decide to divide a big group of devices into two smaller groups and run two FUOTA sessions in parallel instead of running one FUOTA session. The question that arises is whether this decision would make the update any more efficient?

Although the LoRa Alliance tries to standardize the FUOTA process by defining the new specifications, the FUOTA routine itself is not standardized and open for contributions. In this section, a straightforward FUOTA routine that does not convey any kind of smartness is presented. Nevertheless, the proposed FUOTA routine can help us to study and evaluate the process and define the trade-offs in the system design. This can help LoRaWAN operators understand the impact of their system decisions on the process' efficiency and, thus, can help them to devise smarter FUOTA processes.

Consider a firmware update is scheduled for a LoRaWAN deployment, consisting of class A devices. Consequently, the firmware update server configures the same multicast and fragmentation sessions for all devices. An example of the FUOTA session that uses mutlicast class C is shown in Figure 6.2. It should be noted that the multicast and the fragmentation requests are downlink commands and, thus, uplink messages are needed first in order for the devices to open receive windows. Once these two sessions are set up, every device sends *AppTimeReq* command to ask for clock correction before the firmware update server can set up the start time of the multicast transmissions. The start time has to be sufficiently far in the future to guarantee that all devices have set up the required sessions before sending the multicast transmissions. For the multicast class B group, the corresponding commands are used as shown in §6.2.1.

At the exact declared time, all devices must switch to class C and open a continued receive window based on the configuration (data rate and channel) that has been sent in the *McClassCSessionReq* command. At the same time, the firmware update server schedules the firmware fragments one after another until all the fragments, including the redundant ones, have been sent. Once a device receives enough fragments to reconstruct the firmware image, it reverts back to class A. It should be noted that all transmissions, either uplink or downlink are

governed by the duty cycle limitations of the channel used. Following the complete download, an integrity check is done on the received image (details are out-of-scope) and the image is marked as ready if the check is passed. Next, the old firmware is replaced with the new one (details are out-of-scope), which completes the firmware update.

6.4 Performance Evaluation

To evaluate the FUOTA process, a simulation tool, called FUOTASim is developed on top of FREELoRa simulator (chapter 5) FUOTASim leverages the Simpy package for process-based discrete-event simulation in Python. FUOTASim implements the dual-component log-distance pathloss model from [ASC19]*, which was fitted to real LoRaWAN measurements. FUOTASim also considers the packet error model that was presented in [AZCP20], which draws on a probabilistic reception model based on the signal strength and packet length. In addition to that, FUOTASim adopts some features from FREESim [AZCP20] such as the impact of the imperfect orthogonality of spreading factors and the duty cycle limitations, leading to realistic simulation results. Finally, FUOTASim simulates the FUOTA process as described in §6.3 with varying parameter settings. The settings allow choosing between multicast class C or class B to perform FUOTA. In the case of class B, a parameter to configure the ping slot periodicity is presented. Additionally, FUOTASim allows the use of different data rates, firmware sizes, fragment sizes, and redundant codes to perform FUOTA. The aforementioned features are required for a proper evaluation of the FUOTA process, making FUOTASim a useful tool for the LoRaWAN community [3].

The simulation campaigns consider one gateway that is placed in the middle of a LoRaWAN deployment. Class A LoRaWAN devices are spatially distributed around that gateway in such a way so as to acquire a certain data rate distribution across the deployment. The data rate distribution aimed at is to let 45% of the devices use DR5 (SF7/125KHz), 22% use DR4 (SF8/125KHz), 11% use DR3 (SF9/125KHz), 8% use DR2 (SF10/125KHz), 8% use DR1 (SF11/125KHz), and finally 6% use DR0 (SF12/125KHz). This distribution is obtained from a real LoRaWAN deployment in Dublin, Ireland, making the results more realistic. Table 6.1 summarizes all the simulation parameters used in the evaluations. Each simulated study is executed 10 times using different random seeds and the

[3] `https://github.com/kqorany/FUOTASim`

Table 6.1: Simulation Parameters for FUOTA Study

Parameters	**Value [Unit] + Comment**
Random Seeds	10
Devices	100 - 500
Rx1 Window	same configurations as previous uplink
Rx2 Window	DR0 and 869.252 MHz (10% DC)
Data Rate Distribution	[DR0→6%, DR1→8%, DR2→8%, DR3→11%, DR4→22%, DR5→45%]
Gateway Receptions	8 in parallel
LoRaWAN MAC Header	8 [Bytes]
Path Loss [ASC19]*	near ($< 400m, d_0 = 92.67$, $PL_{d_0} = 128.63, \gamma = 1.05, \sigma = 8.72$) far ($\geq 400m, d_0 = 37.27$, $PL_{d_0} = 132.54, \gamma = 0.8, \sigma = 3.34$)
Devices Antenna Gain	2.2 dBm
Gateways Antenna Gain	8 dBm
Application Uplinks	15 [Bytes]
Multicast Transmissions	869.252 MHz (10% DC)
Redundant Fragments	30
Capacity of Batteries	1000 [mAh], 11100 [Joules]
Power Consumption	132 [mW] (Transmission) 48 [mW] (Reception) 0.018 [mW] (Ideal)

mean across all results is presented along with the standard deviation.

The objective of the simulation is to find the best FUOTA parameters regardless of the current LoRaWaN implementations. Therefore, LoRaWAN devices in the simulation are assumed to be equipped with the three LoRaWAN classes (A, B and C) but run normally in class A to minimize the power consumption. In this case, the simulation examines all the possible combinations such as using multicasrt class B and multicast Class C to find the best FUOTA parameters in each scenario. This would give important insights to LoRaWAN developers to include the right functionalities based on the considered scenario. The simulations are divided into two phases: *initial* and *multicast* to show the impact of each phase separately. The initial phase covers everything required before the multicast transmissions can be sent. From Figure 6.2, the initial phase covers the multicast session setup, the fragmentation session setup, the clock synchronization, and the class C session start. However, the multicast phase covers the multicast transmissions. The evaluation results are presented in terms of the following metrics:

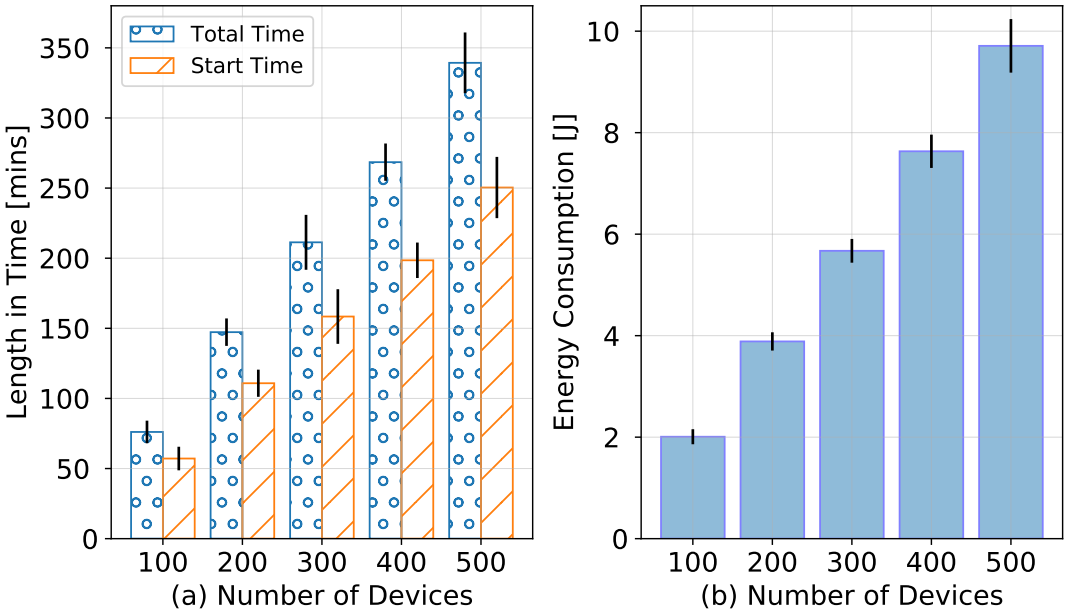


Figure 6.3: Initial Phase - Time Required and Energy Consumption

- *energy consumption*, which indicates the device's average energy consumption.
- *time*, which indicates the average time required to finish a certain task.
- *update efficiency*, which indicates the average ratio of devices that receive the firmware image successfully to the whole deployment.

6.4.1 Initial Phase Study

In this phase, the device's energy consumption (*see* Figure 6.3b) and the total time required (*see* Figure 6.3a (Total Time)) for the devices to go through the initial phase are investigated. Figure 6.3b shows a linear increase in the device's energy consumption over the network size, where the energy consumption increases by approx. 2 Joules with every 100 devices added to the network. A similar trend is observed between the total time and the network size (*see* Figure 6.3a (Total Time)). Figure 6.3a also presents the *start time* metric, which indicates the minimum time required for the devices to complete only the *class C session start* as in Figure 6.2. This time has to be sufficiently far in the future to guarantee that every device receives a *McClassCSessionReq* command and acknowledges the command's receipt. In Figure 6.3a, the start time metric indicates the *earliest time* for the firmware update server to start the multicast

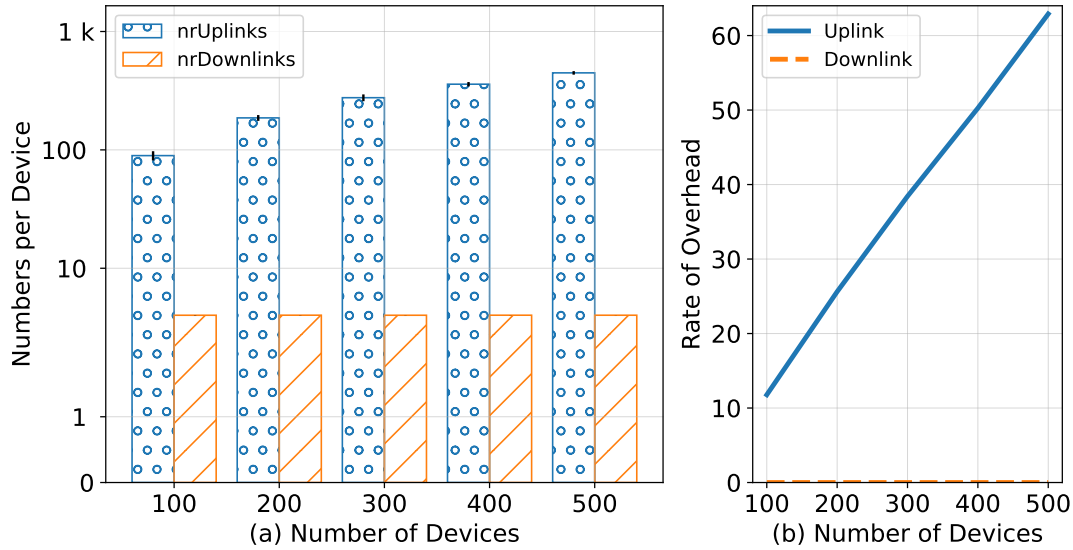


Figure 6.4: Initial Phase - Uplink vs Downlink

transmissions. If the start time is set to be less than the shown results, some devices may miss their *McClassCSessionReq* commands and, thus, miss the multicast transmissions. The metric is shown in minutes, where the reference time is the time of sending the first *McClassCSessionReq* command ever during the initial phase. The linear increase is also observed here between the start time metric and the network size.

End-devices throughout the initial phase rely on the LoRaWAN MAC (i.e. simple Aloha), which is known for its poor scalability, which is even worse when downlink transmissions are required [PRKS17]. This is the main reason behind the linear increasing trend observed in the energy and the time metrics (*see* Figure 6.3). In order to quantify the impact of the scalability issue, Figure 6.4a shows the average number of uplink and downlink transmissions per device throughout the initial phase. In the case of no losses and no duty cycle limitations, only 7 uplink and 4 downlink transmissions are required for each device (*see* Figure 6.2). However, in real conditions, the number of uplink transmissions increases significantly (*see* Figure 6.4a). For instance, in a network with 100 devices, a device sends approx. 94.7 uplink transmissions, which equals approx. 12.5 times more overhead (*see* Figure 6.4b). The overhead rate (Fig. 6.4b) represents the ratio between the numbers in the real condition to the ideal condition. With increasing the network size, the overhead increases linearly for the uplink transmissions (*see* Figure 6.4b). For the downlink transmissions, the losses are very low because of no collisions and the high antenna

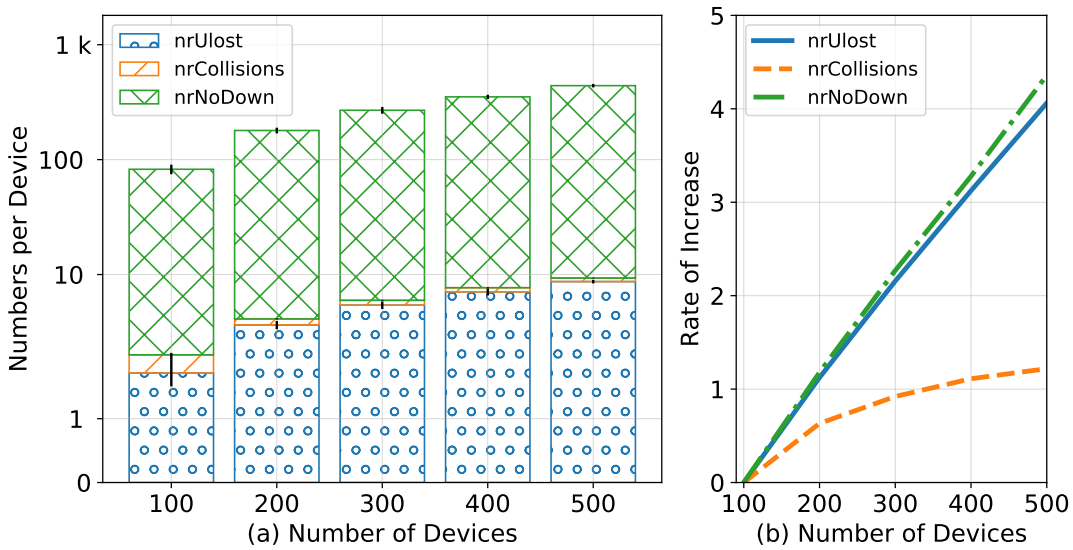


Figure 6.5: Initial Phase - Source of Losses in Uplink Transmissions

gain of the gateways.

Figure 6.5a shows the different sources behind the huge number of uplink transmissions and Figure 6.5b shows the rate of increase in reference to a network size with 100 devices. The rate of increase metric shows the increase from one value to another in terms of a percentage. The main sources of loss are a) collisions (nrCollisions) b) loss due to channel errors (nrUlost) and c) the gateway's duty cycle limitation (nrNoDown). The nrNoDown metric indicates the number of uplink transmissions of a device that have been received correctly by the gateway but the gateway could not transmit the corresponding downlink (in the two receive windows) due to the duty cycle limitation. In this case, a retransmission is scheduled. Surprisingly, the collisions are not the main source of loss as it only presents 0.3% of the losses. This is due to the relatively small network sizes considered in the simulations. For bigger network sizes, the collisions would be a serious source of losses [AZCP20]. The channel fading also has a minimal impact on the losses, about 1.9%. However, the main source of loss is the duty cycle limitation of the gateway (nrNoDown), which presents about 97.8% of the losses. This is a very important conclusion that has to be considered when designing the FUOTA routine. More insights are presented in §6.5.

6.4.2 Multicast Transmissions Phase Study

In this subsection, the impact on time, energy consumption, and efficiency of the varying configurable parameters for the multicast fragments during the firmware update are studied. These parameters include the data rate used, the class of multicast, either C or B, and the ping periodicity in case of class B. Besides these parameters, the impact of the firmware sizes on the aforementioned metrics is studied by considering different sizes: 5kBytes, 10kBytes, 50kBytes, and 100kBytes bytes. As this phase includes only downlink transmissions, the network size does not impact much on the results. Therefore, the presented results are gathered only from a network size with 100 devices from where results can be generalized to the other network sizes.

The fragment size is both directly and inversely proportional to the update time and, thus, the device's energy consumption. On one hand, longer fragment sizes reduce the overall number of fragments, however, the fragments would have a higher probability of error, requiring more redundant fragments. On the other hand, shorter fragment sizes reduce the probability of error but increase the ratio of MAC header (overhead) to payload size, resulting in high overhead. This trade-off has been studied theoretically in section 5.3.1 to compute the best packet/fragment size per data rate. Although the calculations have been done for an uplink use case, its conclusion holds true for the downlink use case as both cases transmit on the same channel. The calculations concluded that the impact of packet errors is not as critical as the impact of the MAC header overhead in terms of time and energy consumption. Therefore, long packets for all data rates are better than short packets to reduce the overall number of transmissions and, thus, the impact of MAC headers.

For this reason, the fragment sizes are set to equal the maximum MAC payload sizes (*see* Table. 2.3). Fig. 6.6 shows the fragment sizes and the airtimes (i.e. transmission times) of one fragment. A clear observation is that the lower the data rate, the higher the airtime even for the same fragment size. For instance, the fragment sizes at DR5 and DR4 are the same but the airtime at DR5 is almost half the airtime at DR4. This is due to the positive relationship between the spreading factor and the airtime [Cor13]. The fragment sizes also determine the number of fragments (*see* Fig. 6.7). These numbers along with the airtimes (*see* Fig. 6.6b) and the duty cycle limitations affect the firmware update time. By increasing the data rate, one would expect a decrease in the update time and the device's energy consumption. Nevertheless, increasing the data rate leads

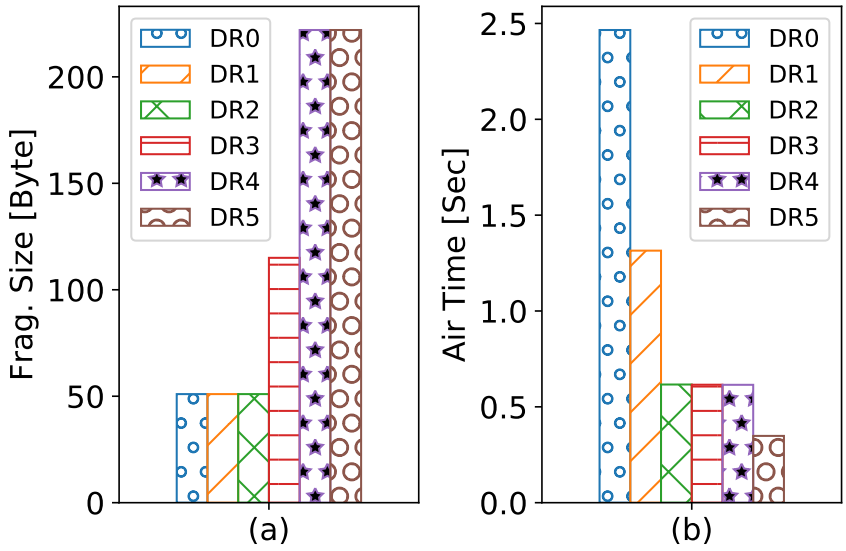


Figure 6.6: Airtime and size of fragments per data rates

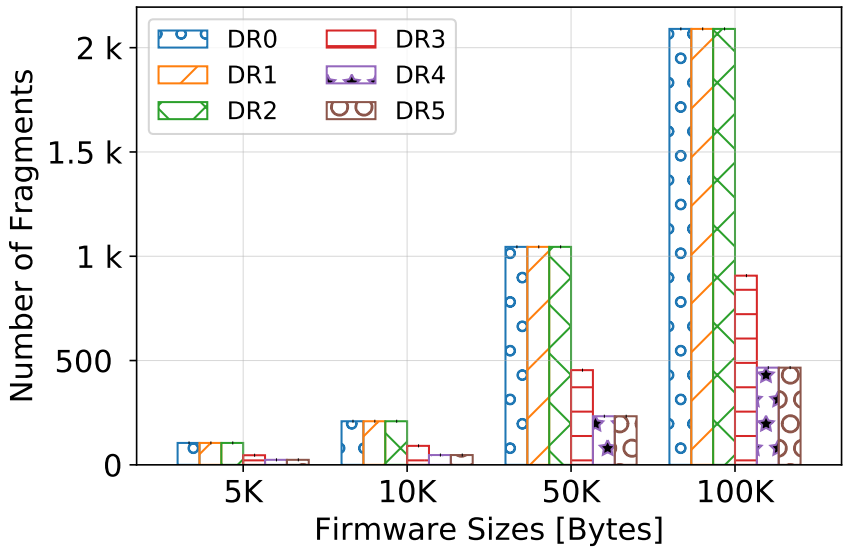


Figure 6.7: Number of Fragments

also to a decrease in the update efficiency as higher data rates have shorter transmission ranges (i.e. lower sensitivity). This trade-off is quantified later in this subsection.

6.4.2.1 Multicast Class C

Figure 6.8a shows the update time across all data rates and for different firmware sizes. Furthermore, Figure 6.8b shows the rate of increase, i.e., percentage increase, metric in terms of the data rates and the firmware sizes in reference to DR0 and firmware size of 5kbytes, respectively. The time metric al-

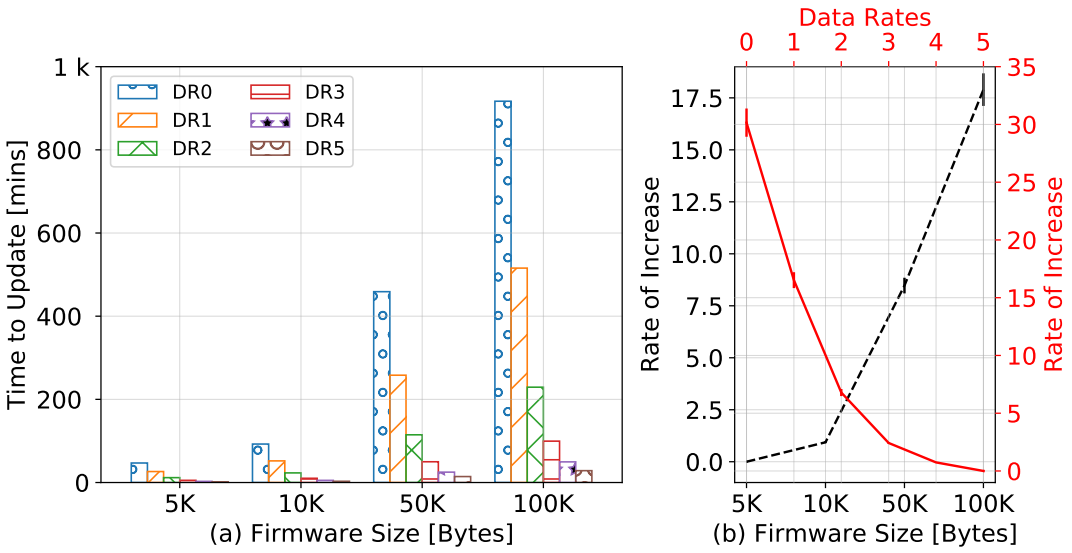


Figure 6.8: Class C - Time to Update

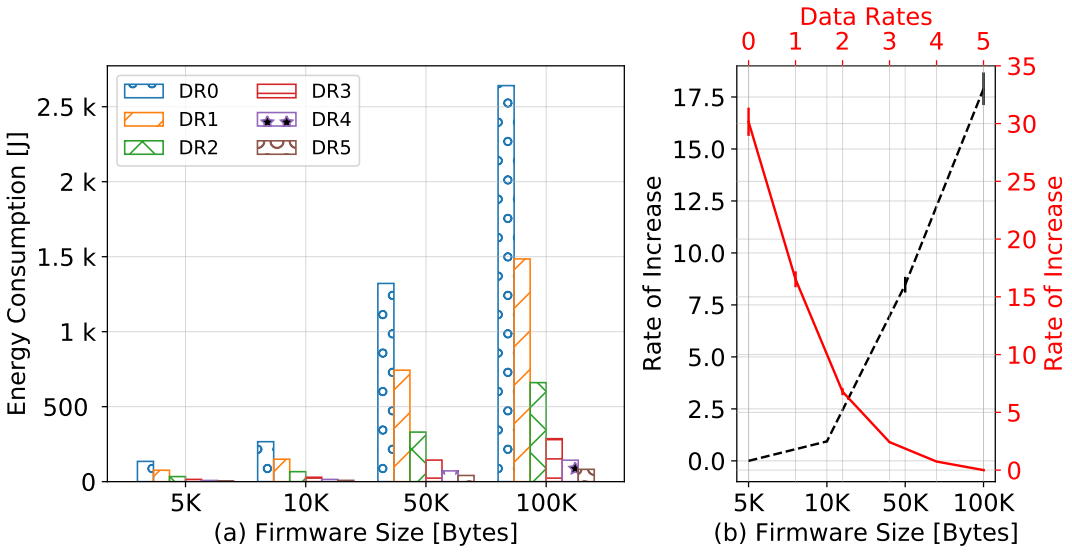


Figure 6.9: Class C - Energy Consumption

most doubles with every time the firmware doubles in size. In addition to that, for the same firmware size, the time metric among the data rates shows the same relationship, showing almost 30 times higher when using DR0 than using DR5. This is mainly due to the large number of fragments and the long air-time in the case of DR0, resulting in long silent periods between two successive transmissions due to the duty cycle of the gateway.

Figure 6.9a shows the device's energy consumption and Figure 6.9b shows the

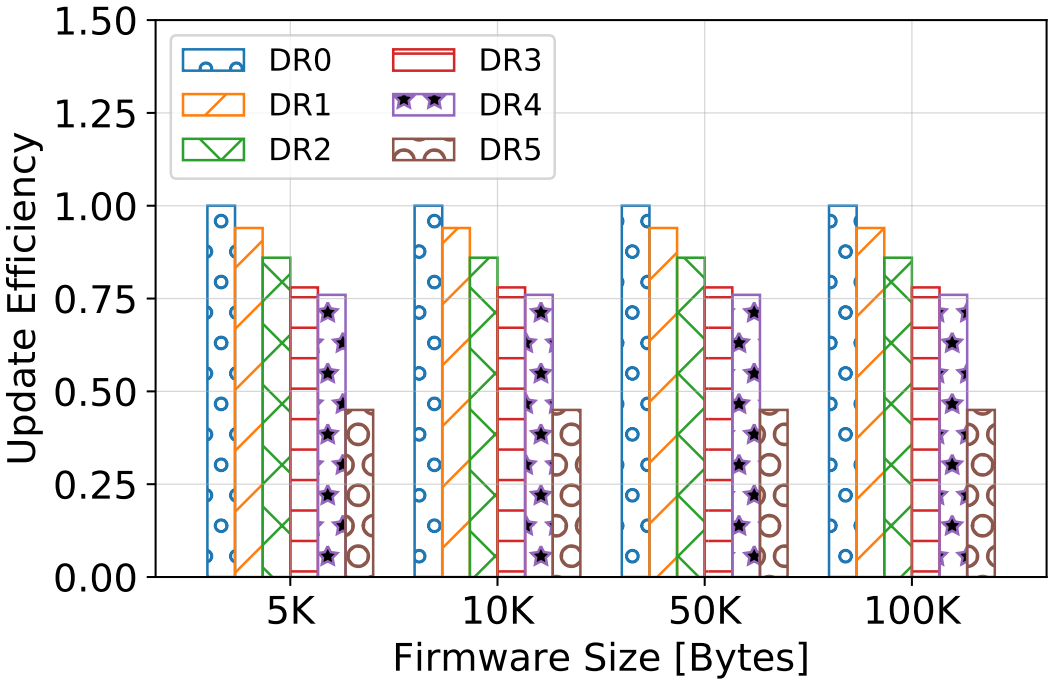


Figure 6.10: Class C - Update Efficiency

rate of increase in terms of the data rates and the firmware sizes. It is obvious that Figure 6.9b is almost identical to Figure 6.8b. This is because energy consumption is proportional to the devices' receiving time. In multicast class C, a device is always in a receive mode for the whole time of the update.

From Figure 6.8 and Figure 6.9 it can be observed that for a certain firmware size, the higher the data rate, the lower the update time and the lower the devices' energy consumption. Nevertheless, another factor has to be considered when choosing the data rate, which is the update efficiency. This is because of the fact that the higher the data rate, the shorter the transmission range and, thus, the lower the update efficiency. Figure 6.10 shows the update efficiency using all data rates and for different firmware sizes. Using DR0, all devices can be updated at once compared to only 45% of the devices in the case of using DR5. The update efficiency metric is directly proportional to the considered data rate distribution (*see* Table 6.1). These results highlight that more than one FUOTA session would be required in the case of using DR5 to update all devices. This would still be acceptable because of the long time and the high energy required in the case of using DR0 (30 times higher than DR5) (*see* Figs. 6.8 and 6.9). However, this is subject to the distribution of gateways in the deployment or their mobility, where a gateway may be able to move to reach

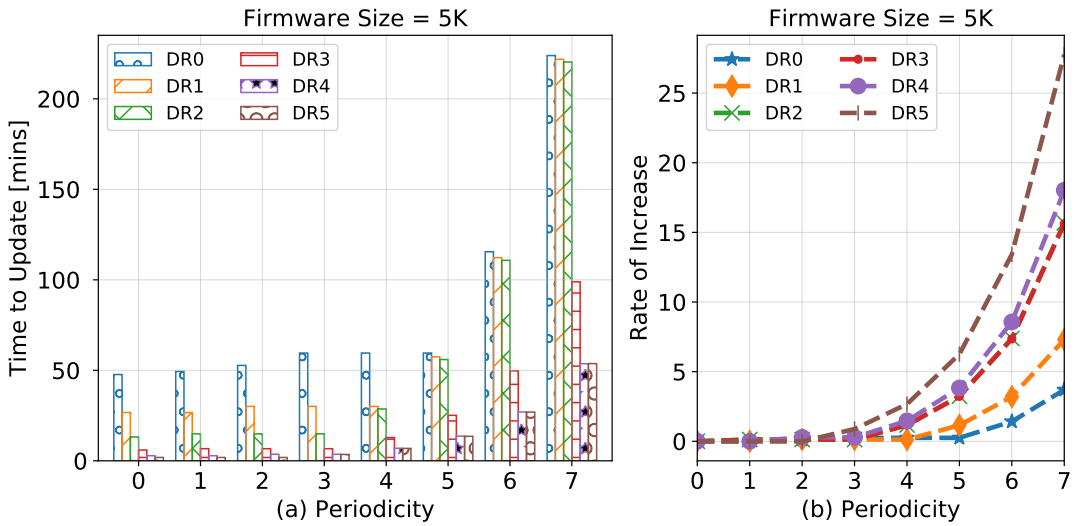


Figure 6.11: Class B - Time to Update

more devices every time, e.g., a drone based gateway.

6.4.2.2 Multicast Class B

The presented results here are from a firmware image of size 5kbytes only and the results of the other sizes can be roughly estimated using the rate of increase from class C (*see* Figure 6.8b). Figure 6.11a shows the update time using all data rates and all ping slot periodicities. Furthermore, Figure 6.11b shows the rate of increase of all data rates in reference to ping periodicity $p = 0$. As shown, $p = 0$ is the best ping periodicity for all data rates as it achieves the lowest update time. This is because of the abundance of ping slots (128 slots) available when $p = 0$, which does not limit the downlink transmissions. The results of $p = 1$ are close enough to the results of $p = 0$. This is because these two ping periodicities are still lower than the duty cycle of the gateway. However, for higher ping periodicities, the time metric increases proportionally to the corresponding ping periodicity and the data rate used.

Figure 6.12a shows the device's energy consumption using all data rates and all ping periodicities. The energy consumption is directly proportional to the devices' receiving time. In class B, the radio of a device is in a receive mode only when receiving downlink fragments, receiving gateway beacons, and checking empty ping slots. Otherwise, the radio of a device is in idle mode. The gateway beacons are received to keep the synchronization with the gateway's clock. Checking empty ping slots happens when a ping slot is assigned for the multi-

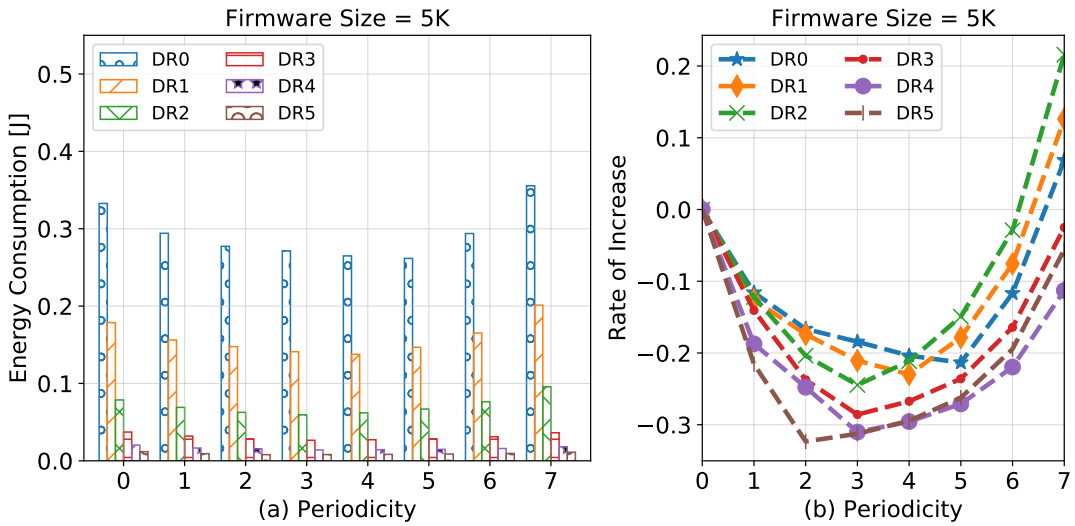


Figure 6.12: Class B - Energy Consumption

cast session but the gateway could not transmit in this slot due to the limited duty cycle. In this case, devices stay in a receive mode at the beginning of the empty slots for the time of a packet preamble. Figure 6.12b shows the rate of increase of all data rates in terms of the device's energy consumption in reference to ping periodicity $p = 0$. It is clear that $p = 0$ is not the best periodicity anymore. However, the energy consumption decreases with increasing the periodicity until a certain periodicity, where afterward the energy consumption starts to increase again. This is because of the relationship between the ping slot periodicity and the gateway's duty cycle. If the periodicity is lower than the gateway's duty cycle, devices check a lot of empty ping slots, resulting in high energy consumption. Also, if the periodicity is higher than the gateway's duty cycle, devices have to receive a lot of gateway beacons to keep synchronization, resulting in high energy consumption as well. The best scenario is to have a periodicity close enough to the gateway's duty cycle. As the gateway's duty cycle depends on the data rate used, the best periodicity varies with the data rate. From Figure 6.12b, $p = 5$ is found to be the best for DR0, $p = 4$ for DR1, $p = 3$ for DR2, $p = 3$ for DR3, $p = 3$ for DR4, and $p = 2$ for DR5.

6.4.2.3 Multicast Class C vs Class B

Fig. 6.13 presents the time to update and the energy consumption of class C and the best configuration of class B (*see* Figs. 6.12). An increase of roughly 17% is observed in the time metric of class B compared to the results of class

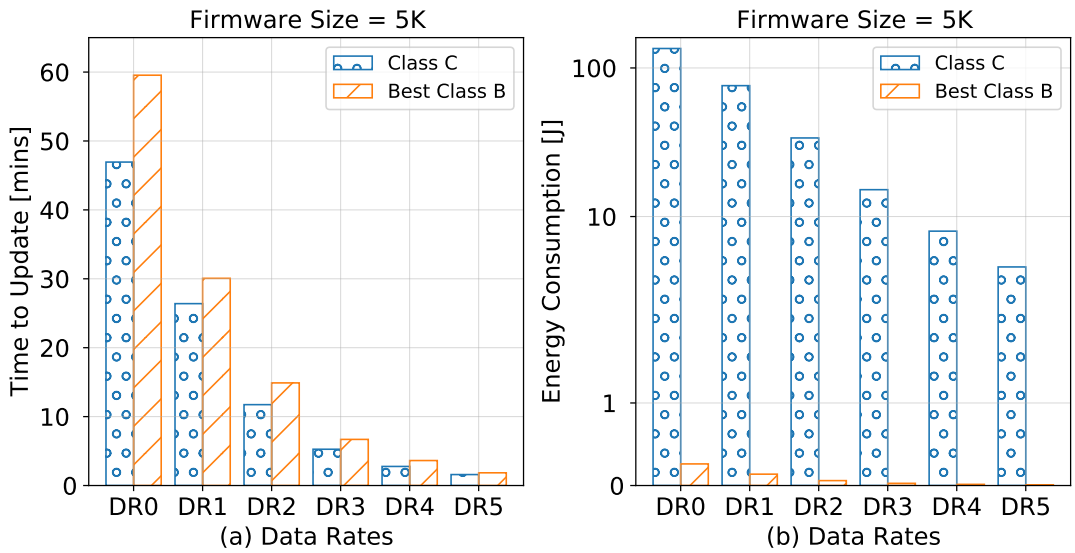


Figure 6.13: Comparison Between Class C and Best of Class B

C. The reason behind this increase is that the downlink transmissions of class B are performed at the beginning of the ping slots only. In this case, even if the duty cycle of the gateway permits to transmit a new downlink fragment, the transmission has to wait until the beginning of the next ping slot, which prolongs the overall update time. In terms of the energy consumption, class B presents a massive reduction in the device's energy consumption up to 550 times less compared to class C.

6.5 Discussion

Here, some ideas are discussed as to how to optimize the FUOTA process for different scenarios. Also, some further insights are provided based on the above simulation results.

6.5.1 Initial Phase

The initial phase is a prerequisite stage every time a firmware update is to be deployed and, therefore, approaches to reduce its overhead in terms of time and energy consumption are desirable. An efficient approach is particularly desirable for frequent small firmware updates such as security patches whereby the overhead of the initial phase can be much higher than transmitting the firmware update itself. The simulation results showed that the main source of the overhead is the duty cycle limitation of the gateway. An approach to overcome this

might be to use multiple co-located gateways, where the overall overhead can be reduced as the network will have higher duty cycle to handle the downlink transmissions as multiple gateways can be used in parallel. The overhead can be also reduced by minimizing the number of transmissions during the initial phase. This can be achieved by combining multiple commands (e.g., multicast setup and fragmentation setup command) in one command and the gateway can acknowledge both commands with just one packet.

Another overhead reduction mechanism might be to pre-program the devices with the required multicast and fragmentation session information before or during deployment into the field as part of the commissioning stage. In this case, the time synchronization and the multicast start will be the only commands required during the initial phase.

6.5.2 Multiple Gateways for Collaborative FUOTA

The rate at which multicast fragments of the firmware can be transmitted is limited by the duty cycle of the gateway. This limitation leads to prolonged firmware update times, which may not be acceptable for certain applications as the normal device operation is blocked during the update time. Consequently, mechanisms to expedite the multicast transmissions are beneficial. Multiple co-located gateways would be a helpful approach here as well, where the gateways transmit the multicast fragments in a collaborative mechanism. In the case of multicast class C, the gateways can transmit the fragments in a round-robin fashion, taking advantage of those devices that are in a continued receive mode. Consequently, the update time is shortened proportional to the number of the gateways used. For example, using ten gateways in a round-robin fashion would achieve an overall 100% duty cycle (each gateway transmits in the 10% duty cycle channel).

In the case of multicast class B, a new multicast session is required every time an additional gateway is used when transmitting the fragments. Here, each gateway handles the ping slots of one multicast session and devices wake up to receive all fragments from all gateways. The drawback here is that this adds additional overhead to the initial phase unless these sessions are pre-programmed into the devices before deployment.

6.5.3 Network Architecture Planning

Using a higher data rate to transmit the firmware fragments is always better in terms of time and energy consumption compared to using a lower data rate. However, in the case where firmware updates are transmitted through only one gateway, the simulation results showed that this negatively affects the update efficiency, where lower data rates achieve better coverage and thus better update efficiency than higher data rates. This trade-off can be settled completely towards high data rates in the case of the multiple gateways scenario. If gateways are geographical distributed such that all devices can be reached when using one of the high data rates, the shortest update time, the lowest energy consummation, and the maximum update efficiency can be achieved all at the same time.

However, network architecture planning is usually not done from a network management point of view such as FUOTA but typically from an application point of view in order to achieve required performance metrics, such as high packet delivery ratio. Nevertheless, network architecture planning based on the requirements for network management would be beneficial when the rate of FUOTA is expected to be high. An example would be where a LoRaWAN deployment is initially based only on a minimal viable application but over time is upgraded to support a wider range of features. LoRaWAN operators take this approach to expedite their position in the market by deploying their network initially with limited applications and relying on FUOTA to extend and optimize applications over time. In this case, FUOTA sessions could be expected to be scheduled more frequently than usual, e.g., once a month.

The cost of deploying enough gateways in order for all devices to be reached using one of the high data rates could be high, especially for very large deployments. In this case, using a mobile gateway could be a reasonable approach. In particular, the deployment is divided geographically to small segments, where if the mobile gateway is deployed in the middle of a segment, the devices within that segment can be reached using the desired data rate. Consequently, the gateway moves from one segment to another to update the devices within each segment. This approach is backed by the simulation results that showed gains in terms of time and energy using similar approaches. For example, running approx. 30 FUOTA sessions using the highest data rate is still more beneficial in terms of energy consumption and update time than running one FUOTA using the lowest data rate.

6.6 Conclusions

In this chapter, the new specifications (multicast, fragmentation, and clock synchronization) by the LoRa Alliance are reviewed, which facilitate efficient FUOTA on top of LoRaWAN. Also, a proposed FUOTA process is presented, showing the impact of the different FUOTA parameters on the performance of the proposed process. The simulation results showed that the initial phase (required for setting up the required sessions) is a bottleneck of the whole process as it does not scale well with increasing network size. The results also showed that multicast class B achieves 17% longer update time and 550 times less energy use compared to multicast class C. Additionally, the simulations demonstrated the significant impact of the data rate used on the overall results. For example, DR5 achieves 30% reduction in update time and device energy consumption compared to DR0. However, this comes at the expense of the firmware update efficiency, where DR5 can only update a portion of the devices (depends on the devices' location distribution) every session whilst DR0 can update all devices at once. The ping slot periodicity p, in case of multicast class B, impacts on the results as well. For all data rates, the fastest firmware update can be done with $p = 0$, however, in terms of the energy consumption, the optimal periodicity varies with varying the data rate. In this case, $p = 5$ is found to be the best for DR0, $p = 4$ for DR1, $p = 3$ for DR2, DR3, and DR4, and $p = 2$ for DR5.

Chapter 7

Conclusions and Future Perspectives

This chapter concludes the doctoral work and discusses the future work that can be pursued on top of this thesis.

7.1 Conclusions

In this section, the most important conclusions from the research presented in this thesis are highlighted. The research focused on selected novel aspects of IP interoperability in LPWANs and resource management in LoRaWAN. Because LPWANs offer different design parameters and design trade-offs e.g., long-transmission range, low-power consumption, constrained duty cycle, etc. than the traditional short-range WSNs, they are a different class of networks. Consequently, the well-known interoperability solutions and resource management protocols from WSNs are not suitable for LPWANs. In this thesis, a number of novel protocols and improvements are proposed to facilitate interoperability between the current protocol design of LPWANs and the Internet as well as to manage the resources of the networks, considering the limitations of the LPWANs such as the limited downlink capability and the constrained duty cycle. Although the thesis focused on LoRaWAN, the conclusions of this thesis can be generalized to other LPWAN technologies because they are sharing the same architecture and limitations as LoRaWAN. The proposals have targeted various layers of the LPWAN protocol stack, from the MAC layer to the application layer. In addition to that, the contributions have been published in a number of

top-tier peer-reviewed IEEE and ACM conferences and journals.

7.1.1 IP Interoperability

In chapter 3, the importance of interoperability via enabling IP connectivity in LPWANs was discussed, which extends the Internet architecture to these networks. Consequently, bringing the power of open networking, standardization, and cooperation to those networks. Typically, header compression is one of the main components to enable IP connectivity, considering that the frame sizes of LPWANs are small. A new IETF mechanism of header compressing, called SCHC was reviewed. The advantage of SCHC is its ability to compress the headers of a full data flow i.e., headers of IPv6, UDP, and CoAP, down to a few bits. SCHC is based on a static context that does not change over time, avoiding complex resynchronization, which is the most resource-consuming operation in any header compression scheme. The performance evaluation of SCHC against 6LoWPAN header compression showed that SCHC achieved almost 70% higher compression efficiency, resulting in lower transmission time and thus longer network lifetime.

The high compression efficiency of SCHC comes at the expense of high memory requirements, specifically, when a large number of SCHC rules are required, which could be an issue for constrained devices. To address this issue, LSCHC was proposed as a new efficient mechanism for storing the rules in device memory without compromising the compression efficiency of SCHC. Instead of composing one context as in SCHC, LSCHC proposed to compose three contexts, where each one carries SCHC rules that target only one layer of the network stack. In this case, LSCHC does not have to waste memory by saving the same rule multiple times. In addition to that, the concept of layer separation of LSCHC reduces complexity and adds flexibility when compressing/decompressing data flows.

Another issue that limits the applicability of SCHC is its static context nature that handles only fixed or pre-programmed data flows. For this, a dummy mapping concept was proposed as a dynamic compression technique that can compress/decompress header fields in unknown data flows. Without the dummy mapping, SCHC sends unknown header fields inline without compression. The idea behind dummy mapping is to link the actual values of unknown header fields to dummy values at the network SCHC compression/decompression unit. Because devices and the network share the same dummy values, only the in-

dices of these values can be used in all inside communication (i.e. between devices and the network). For outside communication (i.e. between the network and the Internet), the network uses the actual real values that have been saved at the network SCHC C/D unit. Each linking between an actual value and a dummy value is valid until a corresponding timer is expired. After the expiration, the same dummy value can be re-linked to a different actual value of a different header field. Consequently, the dummy mapping provides a kind of dynamic context inside the static context to serve dynamic data flows without the need for re-synchronization between the device and the network. Specifically, the dummy mapping suits the limitations of LPWANs, which can help to increase the applicability of SCHC. For evaluating the dummy mapping, an analytical model was derived to quantify its compression efficiency in different configurations. The numerical results, for a LoRaWAN example, showed that the dummy mapping increased the average compression ratio of SCHC by up to 850% for hourly application traffic and up to 575% for minute based application traffic.

7.1.2 Fair Data Extraction Rate in LoRaWAN

In chapter 4, the characteristics of the PHY and the MAC layers of LoRaWAN were discussed. This study revealed that LoRaWAN does not provide a fair data extraction rate for all devices within a LoRaWAN cell, favouring devices close to the gateway and those that use high data rates. The main reasons behind this unfairness lies in the LoRa modulation's *a)* multiple data rates and *b)* capture effect. LoRa modulation provides multiple data rates (corresponding to LoRa parameters, SF, BW and CR), which leads to an unfair collision probability. Typically, devices that use high data rates experience lower collisions than those that use lower data rates. In addition to that, LoRa radio signals experience a capture effect, where the stronger signal can drown the weaker signal at the receiver. Even when the radio signals use different spreading factors, the capture effect can still be observed because the spreading factors do not lead to perfectly orthogonal signals. This imperfect orthogonality leads to about 10% lower fairness in the Jain's index.

In order to address these issues, a novel resource management approach was proposed to achieve fairness in the data extraction rate for all devices regardless of their positions or their data rates. The new approach is called FADR and consists of two algorithms. The first algorithm manages the data rates of the

devices in a way to achieve a fair collision probability for all devices regardless of the network size. This algorithm depends on a derivation of the fairest data rates ratios to be used in a network cell. The second algorithm controls the transmission power levels of the devices in order to balance the received powers within safe margins and thus mitigating the impact of the capture effect. FADR achieves an almost uniform data extraction rate and maintains good device lifetime for all LoRaWAN devices. In comparison with other relevant state-of-art, FADR achieves 12% more fairness in the Jain's index and 8% less power consumption, without scarifying the overall packet delivery ratio.

Nevertheless, the high expected number of nodes in LoRaWAN networks still challenges the achievement of full fairness. This has been noticed in large networks even when deploying the FADR approach with only 60% fairness in the Jain's index. The reason behind this is that LoRaWAN has a limited number of supported transmission power levels (2-14 dBm) and thus a network cannot totally eliminate the impact of the capture effect. Specifically, collisions are typically concentrated in certain areas, which increases the unfairness within LoRaWAN networks. In this case, the only way to enhance the fairness would be to deploy extra gateways.

7.1.3 Reliable and Energy Efficient Data Collections using LoRaWAN

LoRaWAN's promise of wide-area connectivity and low-power consumption make it particularly suited to data collection applications such as environmental monitoring in remote areas. Some of the limits of LoRaWAN deployments for such applications were studied in chapter 5. LoRaWAN was found to experience a high collision rate, specifically, in large network sizes, which limits the network's scalability and reliability. Additionally, the performance gets even worse when using acknowledged transmissions. The main reason behind this problem is the Aloha-based MAC protocol adopted in LoRaWAN. For example, in a common LoRaWAN configuration (spreading factor 12 and 125 kHz bandwidth) the data delivery rate decreases below 50% for gateways serving more than 900 devices. Therefore, standard LoRaWAN is not reliable and not energy-efficient for large-scale data collection applications.

In chapter 5, a novel time-slotted MAC protocol along with a new resource allocation algorithm were proposed to achieve reliable and energy-efficient data collections. The proposal is called FREE. FREE exploits that most environmental

monitoring applications are delay-tolerant. Therefore, FREE proposes to buffer the sensing data at devices and collects it during scheduled bulk transmissions at convenient points in time. This approach can enable efficient scheduled transmissions in a way to eliminate collisions and thus enhances the network's reliability and scalability. The results showed that FREE can achieve more than five-fold increase in devices' lifetime and about 99% network data delivery ratio regardless of the network size and the traffic type.

7.1.4 Firmware Updates over LoRaWAN

In chapter 6, challenges of enabling FUOTA over LPWANs and, in particular, LoRaWAN were discussed. The required features in order to support efficient FUOTA were defined as follows: *(a)* continuous downlinks, *(b)* multicast capability, *(c)* clock synchronization, and *(d)* fragmentation capability. Utilising these new features, the thesis examined how to enable fast and efficient firmware updates over LoRaWAN. Additionally, a full FUOTA process was proposed in order to quantify the impact of the different parameters and to show the trade-offs among them.

The FUOTA process consists of an initial phase and a multicast phase. In the initial phase, the required sessions (multicast, clock synchronization, and fragmentation) are set up at each device individually. While in the multicast phase, downlink transmissions are sent to all *selected* devices at once i.e., multicast. The results showed that the initial phase represents a bottleneck of the whole process as it does not scale well with the network size. The main reason behind this is the limited duty cycle of the gateway. In order to lower this overhead of the initial phase, the multicast and fragmentation sessions can be moved to be a part of a commissioning stage before deploying the devices. Another way to lower the overhead is to use multiple gateways in parallel and thus increase the duty cycle in the downlink of the network.

In the chapter, multicast using LoRa class C is compared to multicast with class B in sending downlink transmissions. The simulation results showed that while multicast class B requires 17% longer update time it results in 550 times less energy use compared to multicast class C. Additionally, the simulations demonstrated the significant impact of the data rate used on the overall results. For example, DR5 achieves 30% reduction in update time and device energy consumption compared to DR0. However, this comes at the expense of the firmware update efficiency, where DR5 can only update a portion of the devices

(depends on the devices' distribution around the gateway) every session whilst DR0 can update all devices within reach of the gateway at once. The ping slot periodicity p, in case of multicast class B, impacts on the results as well. For all data rates, the fastest firmware update can be done with $p = 0$, however, in terms of the energy consumption, the optimal periodicity varies with varying the data rate. In this case, $p = 5$ is found to be the best for DR0, $p = 4$ for DR1, $p = 3$ for DR2, DR3, and DR4, and $p = 2$ for DR5.

7.2 Future Work

This thesis extended the knowledge of LPWANs and more specifically of LoRaWAN. This thesis also provided novel approaches and mechanisms to improve the performance and to extend the applicability of LoRaWAN in the IoT world. In addition to that, the contributions of the thesis have potential implications to be formulated in new standards (e.g., LSCHC and Dummy Mapping), to improve deployments (e.g., FADR),and to facilitate industry applications (e.g., FREE and FUOTA). However, the thesis heavily focused on LoRaWAN as a well-known example of LPWAN technologies and thus it would be interesting to examine the impacts of the proposed protocols and resource management approaches on different LPWAN technologies. Moreover, the thesis lacks real world evaluation of the solutions and methods proposed. Therefore, testing under real world conditions is proposed as future work. For the research presented in this thesis, some future work can still be considered, which is discussed in the following subsections.

7.2.1 IP Interoprability

This thesis focused only on the header compression problem as one of the main components of enabling IP connectivity to LPWANs. However, in order to handle long IP packets, fragmentation capabilities are needed as well. SCHC has proposed a technology agnostic fragmentation framework. This framework has been discussed in some of the recent works [MKH+19, SGGMSI+20, VMP19, AMT+19, SVA18, ANH+19]. These works have evaluated the impact of fragment sizes and different modes of acknowledgements on the fragmentation performance in terms of reliability and energy consumption. However, these works have only studied the performance in a single link scenario, not considering the impact of the network's scalability. Consequently, evaluating the

impact of transmission errors and collisions on the performance of the SCHC fragmentation modes is highly recommended as future work.

In the same context, a novel mechanism of performing fragmentation in LoRaWAN is studied in chapter 6 as part of supporting FUOTA. This fragmentation solution is also technology agnostic and thus can be used for any LPWAN technology. In addition to that, it supports a forward error correction code in which the devices can recover from fragment drops without requesting re-transmission of lost fragments. This is a fundamentally different way of performing fragmentation than in SCHC. Therefore, a thorough comparison between these two approaches is is suggested as future work in order to identify which is better in what situation.

7.2.2 Fair Data Extraction Rate in LoRaWAN

FADR was proposed to achieve a fair data extraction rate for LoRaWAN devices. FADR has only considered a single gateway setup. However, in a typical scenario, multiple gateways would be available in the same geographical area. In order to extend FADR, the study of a multiple gateway scenario is proposed for future work. A starting point could be to study the resource management in terms of data rates (spreading factors and bandwidths) and transmission power levels so as to equalize the traffic load at each gateway in order to achieve a fair data extraction rate to all devices simultaneously. Only few recent work started to look at this problem [CCB+18, PGSDF20]. However, these works do not include the dynamic reconfiguration of LoRaWAN parameters. Consequently, this could be a good extension. In addition to that, it is not clear how the message exchange (needed to manage the resources) between the devices and the gateways in a multiple gateway scenario should be performed. This is critical in dense networks as without coordination between the gateways the downlink channel can be easily saturated, which affects the reliability of any resource management approach badly.

Generally speaking, the resource management problem in LoRaWAN has been studied quite extensively in the literature. Most of the published work in this domain proposed heuristic approaches with different objectives to manage the resources in LoRa-based networks [LRK18, CGM+18b, CV19, BCGT19, SRHGSGH20, HSMSA20]. Only few recent works looked at the optimal solution of the resource management problem [PGSDF20, SQN18]. All published work in this domain have assumed an optimistic scenario with no external inter-

ference. Therefore, a highly recommended future work might be to investigate the impact of interference from other LoRa networks and from other LPWANs on the resource management problem.

7.2.3 Reliable and Energy Efficient Data Collections using LoRaWAN

FREE was proposed to enable fast, reliable and energy-efficient data collections in LoRaWAN. However, due to the imposed duty cycle restriction on LoRawAN devices, the data collection time is highly affected, especially, in small network deployments. Therefore, mechanisms to expedite data collection are worth exploring, including the use of multiple gateways simultaneously. In this case, collaboration between gateways is required to compute the schedule and disseminate it while still achieving zero collisions. In this thesis, minimizing the overall energy consumption was the only focus, however, it would be interesting to examine the scenario of increasing the fairness among devices in terms of the consumed energy. This would require different allocation techniques to exploit the history of the allocations in the previous data collections. This scenario becomes even more interesting when considering the trajectory of a moving gateway, e.g., a gateway on a drone in case of a data mule use case.

Furthermore, the data collection time could be expedited by using listen-before-talk-based MACs, i.e., Carrier Sense Multiple Access (CSMA) [KRP+20, NFU20, TD18a, Pha18a, Pha18b, KRP18, TD18b] or to explore LoRa in the 2.4 GHz band [PM20]. Both approaches can get rid of the duty cycle regulations and thus would help to speed up the data collections. However, in the case of CSMA, this might come at the expense of higher energy consumption as the devices have to perform listening, which is known to be a power-consuming activity. While in the case of using LoRa in 2.4GHz this would come at the expenses of lower transmission range and higher energy consumption.

7.2.4 Firmware Updates on top of LoRaWAN

The FUOTA process in chapter 6 has considered only a single gateway to send the multicast fragments of the firmware. As future work, considering multiple cooperative gateways is worth exploring. This approach would help to decrease the overhead impact of the initial phases as well as expediting the firmware update time, resulting in lower energy consumption. However, new challenges

might arise such as the cooperation among these gateways in light of possible downlink collisions, in particular for devices that might be covered by multiple gateways.

One interesting point for future work is studying how network architecture planning (e.g. gateways locations) impacts firmware update efficiency. This is different to the usual network planning that is done from an application perspective [OG19, ZSH+19]. There is currently almost no published work on this topic.

Furthermore, the use of mobile gateways to achieve fast and reliable firmware updates deserves further exploring. This approach is backed by the simulation results (Chapter 6) that showed gains in terms of time and energy using similar approaches. For example, running approx. 30 FUOTA sessions using the highest data rate is still more beneficial in terms of update time and energy consumption than running one FUOTA using the lowest data rate. However, it is not clear now to plan the trajectory of the mobile gateway, considering the constraints such as the deployment area and the power bank of the mobile gateway, etc.

Appendix A

The Fair SF Ratios Derivation

LoRaWAN's MAC uses a simple Aloha-style MAC. Consequently, the probability of having at least one collision with the same Data Rate (DR) d can be written as follows:

$$P_d^{coll} = 1 - e^{-2G_d}, \tag{A.1}$$

where G_d is the amount of transmitted packets using DR d during the transmission of one packet using the same DR.

G_d can be calculated as follows:

$$G_d = \lambda \alpha_d T_d \qquad [\textit{packets/ unit time}], \tag{A.2}$$

where λ is the amount of generated traffic per unit time and equals N/I. N is the total number of devices and I is the average packet interval time per device. α_d denotes the ratio of devices that use DR d that also equals ($\alpha_f \alpha_{f,b} \alpha_{f,b,c}$), where α_f denotes the ratio of devices that use SF f, $\alpha_{f,b}$ denotes the ratio of devices that use BW b from the devices ratio that use SF f, and $\alpha_{f,b,c}$ denotes the ratio of devices that use CR c from the devices ratio that use SF f and BW b. Finally, T_d indicates the transmission time of a packet using DR d. T_d can be represented using Equation A.3, where l is the packet length, b is the BW, and c is the CR.

$$\begin{aligned} T_d &= \frac{l}{R_d}, \quad where \\ R_d &= \frac{fb}{2^{f+1}} c \end{aligned} \tag{A.3}$$

Combining Equations A.1, A.2 and A.3 yields $P_{coll,d}$ as

$$P_d^{coll} = 1 - e^{\frac{2^{f+1}}{f}\frac{l}{cb}\alpha_f\alpha_{f,b}\alpha_{f,b,c}\lambda}. \tag{A.4}$$

In order to achieve fairness, the optimization problem in Equation A.5 has to be solved, where the maximum collision probability of a certain DR has to be minimized.

$$\min \max_d P_d^{coll}. \tag{A.5}$$

This optimization problem is constrained as the sum of all SF ratios should be unity (Equation A.6) because each device uses only one SF. The same holds also for BW and CR as the sum of all BW ratios (Equation A.7) and CR (Equation A.8) ratios should be unity.

$$\sum_f \alpha_f = 1, \tag{A.6}$$

$$\sum_b \alpha_{f,b} = 1, \tag{A.7}$$

$$\sum_c \alpha_{f,b,c} = 1. \tag{A.8}$$

In the case where all devices use the same BW and CR, the above optimization problem results in the following ratios (Equation A.9 [1]), which are the fair ratios of using each SF.

$$\alpha_f = \frac{f}{2^f} / \sum_{i=7}^{12} \frac{i}{2^i}. \tag{A.9}$$

Consequently, in the case of using different BWs but the same CR, the fair ratios result in the following equation (Equation A.10).

$$\alpha_{f,b} = b / \sum_{j=BWs} j, \tag{A.10}$$

where BWs in the equation indicates the BWs used in the network. For example, if 125KHz and 250KHz BWs are used, then $\sum_{j=BWs} j = 375000$. In a similar manner, Equation A.11 shows the impact of using different CRs.

$$\alpha_{f,b,c} = c / \sum_{k=CRs} k. \tag{A.11}$$

Combining Equations A.9, A.10, and A.11 yields Equation A.12, which shows

[1] This equation was first presented in [RMP17]

Table A.1: Fair Data Rate Ratios

Data Rates	Configurations	Fair Ratios
0	SF12/BW125KHz/CR1/2	0.024
1	SF11/BW125KHz/CR1/2	0.044
2	SF10/BW125KHz/CR1/2	0.08
3	SF9/BW125KHz/CR1/2	0.144
4	SF8/BW125KHz/CR1/2	0.257
5	SF7/BW125KHz/CR1/2	0.15
6	SF7/BW250KHz/CR1/2	0.3

the fair ratios of deploying each DR.

$$\alpha_d = \frac{fbc}{2^f} / (\sum_{i=7}^{12} \frac{i}{2^i} \sum_{j=BWs} j \sum_{k=CRs} k) \tag{A.12}$$

In LoRaWAN, the SF f ranges from 7 to 12. The BW b options are 125KHz, and 250KHz, and the typical CR c value used is 1/2. In this case, Table A.1 shows the fair ratios of deploying each DR based on Equation A.12.

Appendix B

Structure of Messages Used in FREE

Size (byte)	8	8	2	3	3
Join-request	JoinEUI	DevEUI	DevNone	DataSize	DElasticity

(a) Join-request Message

Size (byte)	3	4	1	1	7
Join-accept	NetID	DevAddr	DLSettings	RxDelay	DCSettings

(b) Join-accept Message

Size (bit)	[55:52]	[51:48]	[47:32]	[31:16]	[15:0]
DCSettings	DataRate	TxPower	ChMask	SlotFrame	SecondStage

(c) DCSettings Field Format

Size (byte)	6	6	6	2	3
FSettings	PcktSizes	Guards	FrameLens	DataCollection	NextRound

(d) FSettings Message

Figure B.1: Fields of Join-request, Join-accept, and FSettings messages

Figure B.1 shows the message structures of *join-request*, *join-accept*, and *FSettings* messages, which are used in the joining and synchronization phase as illustrated in Section 5.2.2. The first three fields of the *join-request* message have the same meaning as in a standard LoRaWAN *join-request* message. Similarly, the first four fields of the *join-accept* message have the same meaning as in a standard LoRaWAN *join-accept* message. However, the *DCSetting* field is added to let the gateway control the transmission parameters of each device. The for-

mat of this field is depicted in Figure B.1c, which consists of *DataRate*, *TxPower*, *ChMask*, *SlotFrame*, and *Secondstage* subfields. The first three subfields are used to control the data rate, transmission output power, and the uplink channels and are following the same format as in the LoRaWAN *LinkADRReq* Mac Command. Finally, *PcktSizes*, *Guards* and *FrameLends* of the *FSettings* message are decoded in a way where the most significant byte in each field corresponds to spreading factor 7, the following byte to spreading factor 8 and so on.

References

[80206] IEEE std. 802.15.4. Part 15.4: wireless medium access control (mac) and physical layer (phy) specifications for low-rate wireless personal area networks (lr-wpans). *IEEE standard for Information Technology*, 2006.

[A+09] Kevin Ashton et al. That 'internet of things' thing. *RFID journal*, 22(7):97–114, 2009.

[ABB+14] Toni Adame, Albert Bel, Boris Bellalta, Jaume Barcelo, and Miquel Oliver. Ieee 802.11 ah: the wifi approach for m2m communications. *IEEE Wireless Communications*, 21(6):144–152, 2014.

[ABIK06] Bassam Aoun, Raouf Boutaba, Youssef Iraqi, and Gary Kenward. Gateway placement optimization in wireless mesh networks with qos constraints. *IEEE Journal on Selected Areas in Communications*, 24(11):2127–2136, 2006.

[ACC+08] Edoardo Amaldi, Antonio Capone, Matteo Cesana, Ilario Filippini, and Federico Malucelli. Optimization models and methods for planning wireless mesh networks. *Computer Networks*, 52(11):2159–2171, 2008.

[ACP17] Khaled Q Abdelfadeel, Victor Cionca, and Dirk Pesch. Lschc: Layered static context header compression for lpwans. In *Proceedings of the 12th Workshop on Challenged Networks*, pages 13–18. ACM, 2017.

[ACP18] Khaled Q. Abdelfadeel, Victor Cionca, and Dirk Pesch. Poster: A Fair Adaptive Data Rate Algorithm for LoRaWAN. In *International Conference on Embedded Wireless Systems and Networks*, EWSN '18. ACM, 2018.

[AGP+15] Sergey Andreev, Olga Galinina, Alexander Pyattaev, Mikhail Gerasimenko, Tuomas Tirronen, Johan Torsner, Joachim Sachs, Mischa Dohler, and Yevgeni Koucheryavy. Understanding the iot connectivity landscape: a contemporary m2m radio technology roadmap. *IEEE Communications Magazine*, 53(9):32–40, 2015.

[AIM10] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer networks*, 54(15):2787–2805, 2010.

[AIM17] Luigi Atzori, Antonio Iera, and Giacomo Morabito. Understanding the internet of things: definition, potentials, and societal role of a fast evolving paradigm. *Ad Hoc Networks*, 56:122–140, 2017.

[AMT+19] Sergio Aguilar, Alexandre Marquet, Laurent Toutain, Carles Gomez, Rafael Vidal, Nicolas Montavont, and Georgios Z Papadopoulos. Lorawan schc fragmentation demystified. In *International Conference on Ad-Hoc Networks and Wireless*, pages 213–227. Springer, 2019.

[Ang01] Ian Angus. An introduction to erlang b and erlang c. *Telemanagement*, 187:6–8, 2001.

[ANH+19] Wael Ayoub, Fabienne Nouvel, Sarah Hmede, Abed Ellatif Samhat, Mohamad Mroue, and Jean-Christophe Prévotet. Implementation of schc in ns-3 and comparison with 6lowpan. In *2019 26th International Conference on Telecommunications (ICT)*, pages 432–436. IEEE, 2019.

[AP19] Andrea Abrardo and Alessandro Pozzebon. A multi-hop lora linear sensor network for the monitoring of underground environments: the case of the medieval aqueducts in siena, italy. *Sensors*, 19(2):402, 2019.

[AR01] Ivo Adan and Jacques Resing. *Queueing Theory: Ivo Adan and Jacques Resing*. Eindhoven University of Technology. Department of Mathematics and Computing Science, 2001.

[AS14] Mohamed Aref and Axel Sikora. Free space range measurements with semtech lora technology. In *IDAACS*, pages 19–23,

Sept 2014.

[ASC19] K. Q. Abdelfadeel, Y. Samarawickrama, and V. Cionca. How to conduct lorawan site surveys. In *2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 133–138, Oct 2019.

[AVTP+17] Ferran Adelantado, Xavier Vilajosana, Pere Tuset-Peiro, Borja Martinez, Joan Melia-Segui, and Thomas Watteyne. Understanding the limits of lorawan. *IEEE Communications Magazine*, 55(9):34–40, 2017.

[AZCP20] K. Q. Abdelfadeel, D. Zorbas, V. Cionca, and D. Pesch. $free$ —fine-grained scheduling for reliable and energy-efficient data collection in lorawan. *IEEE Internet of Things Journal*, 7(1):669–683, Jan. 2020.

[Bag05] Aline Baggio. Wireless Sensor Networks in Precision Agriculture. In *ACM Workshop on Real-World Wireless Sensor Networks (REALWSN 2005), Stockholm, Sweden*, volume 20. ACM, 2005.

[BCGT19] Giuseppe Bianchi, Francesca Cuomo, Domenico Garlisi, and Ilenia Tinnirello. Sequential waterfilling for adaptive data rate allocation in lorawan. *arXiv preprint arXiv:1907.12360*, 2019.

[BDF+19] F Bonafini, A Depari, P Ferrari, A Flammini, M Pasetti, S Rinaldi, E Sisinni, and M Gidlund. Exploiting localization systems for LoRaWAN transmission scheduling in industrial applications. In *15th IEEE International Workshop on Factory Communication Systems (WFCS)*, pages 1–8. IEEE, 2019.

[BH13] Carsten Bormann and Paul E. Hoffman. Concise Binary Object Representation (CBOR). RFC 7049, October 2013.

[BHG11] Djohara Benyamina, Abdelhakim Hafid, and Michel Gendreau. Wireless mesh networks design—a survey. *IEEE Communications surveys & tutorials*, 14(2):299–310, 2011.

[BKKI20] Ben Buurman, Joarder Kamruzzaman, Gour Karmakar, and Syed Islam. Low-power wide-area networks: Design goals, architecture, suitability to use cases and research challenges. *IEEE Access*, 8:17179–17220, 2020.

[BKL17] Dmitry Bankov, Evgeny Khorov, and Andrey Lyakhov. Mathematical Model of LoRaWAN Channel Access. In *WoWMoM*, pages 1–3, June 2017.

[Blu10] SIG Bluetooth. Bluetooth core specification version 4.0. *Specification of the Bluetooth System*, 1:7, 2010.

[Blu17] SIG Bluetooth. Ble mesh profile specification v1. 0, 2017.

[BMOG20] Luca Beltramelli, Aamir Mahmood, Patrik Osterberg, and Mikael Gidlund. Lora beyond aloha: An investigation of alternative random access protocols. *IEEE Transactions on Industrial Informatics*, 2020.

[Bor14] Carsten Bormann. 6lowpan-ghc: generic header compression for ipv6 over low-power wireless personal area networks (6lowpans). *Internet Engineering Task Force (IETF), RFC*, 7400:1–24, 2014.

[BR17] Martin Bor and Utz Roedig. Lora transmission parameter selection. In *DCOSS*, pages 5–7, 2017.

[BRSH18] Mathias Baert, Jen Rossey, Adnan Shahid, and Jeroen Hoebeke. The bluetooth mesh standard: An overview and experimental evaluation. *Sensors*, 18(8):2409, 2018.

[BRVA16] Martin C Bor, Utz Roedig, Thiemo Voigt, and Juan M Alonso. Do LoRa Low-power Wide-area Networks Scale? In *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, pages 59–67. ACM, 2016.

[BS06] Stephen Brown and Cormac Sreenan. Updating software in wireless sensor networks: A survey. *Dept. of Computer Science, National Univ. of Ireland, Maynooth, Tech. Rep*, pages 1–14, 2006.

[CACP] D. Zorbas C. Caillouet, K. Q. Abdelfadeel, V. Cionca, and D. Pesch. Optimal data collection time in lora networks: a time-slotted approach. *IEEE Internet of Things Journal*. Under review.

[CCB+18] Francesca Cuomo, Manuel Campo, Enrico Bassetti, Lorenzo Cartella, Federica Sole, and Giuseppe Bianchi. Adaptive mitigation of the air-time pressure in lora multi-gateway architectures. In *European Wireless 2018; 24th European Wireless Conference*, pages 1–6. VDE, 2018.

[CCC+17] Francesca Cuomo, Manuel Campo, Alberto Caponi, Giuseppe Bianchi, Giampaolo Rossini, and Patrizio Pisani. Explora: Extending the performance of lora by suitable spreading factor allocations. In *WiMob*, pages 1–8, Oct 2017.

[CCG+20] Lin-Heng Chang, Yi Chang, Chih-Kae Guan, Tong-Ying Juang, and Wen-Chang Fang. An adaptive data rate algorithm for improving energy efficiency for multi-gateway lorawans. *International Journal of Ad Hoc and Ubiquitous Computing*, 33(4):197–215, 2020.

[CGM+18a] D. Croce, M. Gucciardo, S. Mangione, G. Santaromita, and I. Tinnirello. Impact of LoRa Imperfect Orthogonality: Analysis of Link-Level Performance. *IEEE Communications Letters*, 22(4):796–799, April 2018.

[CGM+18b] Francesca Cuomo, Julio César Carrasquel Gámez, Antonio Maurizio, Laura Scipione, Manuel Campo, Alberto Caponi, Giuseppe Bianchi, Giampaolo Rossini, and Patrizio Pisani. Towards traffic-oriented spreading factor allocations in lorawan systems. In *2018 17th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, pages 1–8. IEEE, 2018.

[CGT+17] Daniele Croce, Michele Gucciardo, Ilenia Tinnirello, Domenico Garlisi, and Stefano Mangione. Impact of Spreading Factor Imperfect Orthogonality in LoRa Communications. In *International Tyrrhenian Workshop on Digital Communication*, pages 165–179. Springer, 2017.

[Cis17] V Cisco. The zettabyte era: trends and analysis. updated (07/06/2017), 2017.

[CMVG17] Lluís Casals, Bernat Mir, Rafael Vidal, and Carles Gomez. Modeling the Energy Performance of LoRaWAN. *Sensors*, 17(10), 2017.

[Cor13] Semtech Corporation. Lora modem design guide. `$semtech.com/images/datasheet/LoraDesignGuide_STD.pdf$`, 2013. Online; accessed 10-December-2017.

[CV19] Marco Centenaro and Lorenzo Vangelista. Time-power multiplexing for lora-based iot networks: An effective way to boost lorawan network capacity. *International Journal of Wireless Information Networks*, 26(4):308–318, 2019.

[DCABM05] Douglas SJ De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A High-throughput Path Metric for Multi-hop Wireless Routing. *Wireless networks*, 11(4):419–434, 2005.

[DEA06] Ilker Demirkol, Cem Ersoy, and Fatih Alagoz. MAC Protocols for Wireless Sensor Networks: A Survey. *IEEE Communications Magazine*, 44(4):115–121, 2006.

[DFDA11] Mario Di Francesco, Sajal K Das, and Giuseppe Anastasi. Data Collection in Wireless Sensor Networks with Mobile Elements: A Survey. *ACM Transactions on Sensor Networks (TOSN)*, 8(1):7, 2011.

[DZGPRPM16] Almudena Díaz-Zayas, Cesar A García-Pérez, Alvaro M Recio-Pérez, and Pedro Merino. 3gpp standards to deliver lte connectivity for iot. In *2016 IEEE first international conference on internet-of-things design and implementation (IoTDI)*, pages 283–288. IEEE, 2016.

[Eri16] A Ericsson. Cellular networks for massive iot—enabling low power wide area applications. *no. January*, pages 1–13, 2016.

[ESRB19] C. Ebi, F. Schaltegger, A. Rüst, and F. Blumensaat. Synchronous LoRa Mesh Network to Monitor Processes in Underground Infrastructure. *IEEE Access*, 7:57663–57677, Sep 2019.

[ETS12] ETSI ERM TG28. Electromagnetic Compatibility and Radio Spectrum Matters (ERM); Short Range Devices (SRD); Radio equipment to be used in the 25 MHz to 1 000 MHz Frequency Range with Power Levels Ranging up to 500 mW, 2012. V2.4.1.

[Far18] Stephen Farrell. Low-Power Wide Area Network (LPWAN) Overview. RFC 8376, May 2018.

[FRP17] Dustin Frisch, Sven Reißmann, and Christian Pape. 'an over the air update mechanism for esp8266 microcontrollers. In *Proc. 12th Int. Conf. Syst. Netw. Commun.(ICSNC)*, 2017.

[FUO18] FUOTA Working Group of the LoRa Alliance Technical Committee. LoRaWAN Fragmented Data Block Transport Specification. `https://lora-alliance.org/sites/default/files/2018-09/fragmented_data_block_transport_v1.0.0.pdf`, 2018. Accessed: 2-January-2020.

[FUO19] FUOTA Working Group of the LoRa Alliance Technical Committee. FUOTA Process Summary. `https://lora-alliance.org/sites/default/files/2019-04/tr002-fuota_process_summary-v1.0.0.pdf`, 2019. Accessed: 2-January-2020.

[GdSCK17] Yasaman Ghasempour, Claudio RCM da Silva, Carlos Cordeiro, and Edward W Knightly. Ieee 802.11 ay: Next-generation 60 ghz communication for 100 gb/s wi-fi. *IEEE Communications Magazine*, 55(12):186–192, 2017.

[GDZ+19] Weifeng Gao, Wan Du, Zhiwei Zhao, Geyong Min, and Mukesh Singhal. Towards energy-fairness in lora networks. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pages 788–798. IEEE, 2019.

[GG15] Claire Goursaud and Jean-Marie Gorce. Dedicated networks for iot: Phy/mac state of the art and challenges. *EAI endorsed transactions on Internet of Things*, 2015.

[GOP12] Carles Gomez, Joaquim Oller, and Josep Paradells. Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology. *Sensors*, 12(9):11734–11753, 2012.

[GPBC17] C. Gomez, J. Paradells, C. Bormann, and J. Crowcroft. From 6lowpan to 6lo: Expanding the universe of ipv6-supported technologies for the internet of things. *IEEE Communications Magazine*, 55(12):148–155, DECEMBER 2017.

[GR17] Orestis Georgiou and Usman Raza. Low power wide area network analysis: Can lora scale? *IEEE Wireless Communications*

Letters, 6(2):162–165, 2017.

[GVGZ93] Sudheer A Grandhi, Rajiv Vijayan, David J Goodman, and Jens Zander. Centralized power control in cellular radio systems. *IEEE Transactions on Vehicular Technology*, 42(4):466–468, 1993.

[GVN16] Lukas Gregora, Lukas Vojtech, and Marek Neruda. Indoor signal propagation of lora technology. In *2016 17th International Conference on Mechatronics-Mechatronika (ME)*, pages 1–4. IEEE, 2016.

[GYYL09] Khusvinder Gill, Shuang-Hua Yang, Fang Yao, and Xin Lu. A zigbee-based home automation system. *IEEE Transactions on consumer Electronics*, 55(2):422–430, 2009.

[GZDJ19] S. Gao, X. Zhang, C. Du, and Q. Ji. A Multichannel Low-Power Wide-Area Network With High-Accuracy Synchronization Ability for Machine Vibration Monitoring. *IEEE Internet of Things Journal*, 6(3):5040–5047, June 2019.

[GZG12] Jeremy Gummeson, Pengyu Zhang, and Deepak Ganesan. Flit: A Bulk Transmission Protocol for RFID-scale Sensors. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 71–84. ACM, 2012.

[H+17] Niklas Heuveldop et al. Ericsson mobility report. *Ericsson AB, Technol. Emerg. Business, Stockholm, Sweden, Tech. Rep. EAB-17*, 5964, 2017.

[HC04] Jonathan W Hui and David Culler. The dynamic behavior of a data dissemination protocol for network programming at scale. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 81–94. ACM, 2004.

[HDPMH18] Jetmir Haxhibeqiri, Eli De Poorter, Ingrid Moerman, and Jeroen Hoebeke. A Survey of LoRaWAN for IoT: From Technology to Application. *Sensors*, 18(11), 2018.

[HJY+20] Quy Lam Hoang, Woo-Sung Jung, Taehyun Yoon, Daeseung Yoo, and Hoon Oh. A real-time lora protocol for industrial monitoring and control systems. *IEEE Access*, 8:44727–44738, 2020.

[HOOF20] Martin Haubro, Charalampos Orfanidis, George Oikonomou, and Xenofon Fafoutis. Tsch-over-lora: Long range and reliable ipv6 multi-hop networks for the internet of things. *Internet Technology Letters*, 2020.

[HOWM14] Victoria J Hodge, Simon O'Keefe, Michael Weeks, and Anthony Moulds. Wireless Sensor Networks for Condition Monitoring in the Railway Industry: A Survey. *IEEE Transactions on Intelligent Transportation Systems*, 16(3):1088–1106, 2014.

[HPON13] SHI Huaizhou, R Venkatesha Prasad, Ertan Onur, and IGMM Niemegeers. Fairness in wireless networks: Issues, measures and challenges. *IEEE Communications Surveys & Tutorials*, 16(1):5–24, 2013.

[HSMSA20] Arliones Hoeller, Richard Demo Souza, Samuel Montejo-Sánchez, and Hirley Alves. Performance analysis of single-cell adaptive data rate-enabled lorawan. *IEEE Wireless Communications Letters*, 2020.

[HTB+08] Andreas Hasler, Igor Talzi, Jan Beutel, Christian Tschudin, and Stephan Gruber. Wireless Sensor Networks in Permafrost Research-concept, Requirements, Implementation and Challenges. In *Proc. 9th International Conference on Permafrost (NICOP 2008)*, volume 1, pages 669–674, 2008.

[Jac12] Eric Jacobsen. Understanding and Relating Eb/No, SNR, and other Power Efficiency Metrics. `www.dsprelated.com/showarticle/168.php`, 2012. Accessed: 2018-01-04.

[JCH84] Raj Jain, Dah-Ming Chiu, and William R Hawe. *A quantitative measure of fairness and discrimination for resource allocation in shared computer system*, volume 38. Eastern Research Laboratory, Digital Equipment Corporation Hudson, MA, 1984.

[JKM+17] Petäjäjärvi Juha, Mikhaylov Konstantin, Pettissalo Marko, Janhunen Janne, and Iinatti Jari. Performance of a Low-power Wide-area Network Based on LoRa Technology: Doppler Robustness, Scalability, and Coverage. *International Journal of Distributed Sensor Networks*, 13(3), 2017.

[Jon18] Jongboom, Jan and Stokking, Johan. Enabling firmware updates over LPWANs. `http://janjongboom.com/downloads/ew2018-paper.pdf`, 2018. Unpublished; Accessed: 2-January-2020.

[KFD+07] Sukun Kim, Rodrigo Fonseca, Prabal Dutta, Arsalan Tavakoli, David Culler, Philip Levis, Scott Shenker, and Ion Stoica. Flush: A Reliable Bulk Transport Protocol for Multihop Wireless Networks. In *SenSys '07*, pages 351–365. ACM, 2007.

[KLKG15] Evgeny Khorov, Andrey Lyakhov, Alexander Krotov, and Andrey Guschin. A survey on ieee 802.11 ah: An enabling networking technology for smart cities. *Computer Communications*, 58:53–69, 2015.

[KRP18] Nikos Kouvelas, Vijay Rao, and RR Prasad. Employing p-csma on a lora network simulator. *arXiv preprint arXiv:1805.12263*, 2018.

[KRP+20] Nikolaos Kouvelas, Vijay S Rao, R Venkatesha Prasad, Gauri Tawde, and Koen Langendoen. p-carma: Politely scaling lorawan. In *International Conference on Embedded Wireless Systems and Networks*, EWSN '20. ACM, 2020.

[L+16] Sam Lucero et al. Iot platforms: enabling the internet of things. *White paper*, 2016.

[LAG+09] Ming Li, Devesh Agrawal, Deepak Ganesan, Arun Venkataramani, and Himanshu Agrawal. Block-switched Networks: A New Paradigm for Wireless Transport. In *NSDI*, volume 9, pages 423–436, 2009.

[LBPB18] Luca Leonardi, Filippo Battaglia, Gaetano Patti, and Lucia Lo Bello. Industrial lora: A novel medium access strategy for lora in industry 4.0 applications. In *IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society*, pages 4141–4146. IEEE, 2018.

[LCH+11] Philip Levis, Thomas Clausen, Jonathan Hui, Omprakash Gnawali, and J Ko. The trickle algorithm. *Internet Engineering Task Force, RFC6206*, 2011.

[LJC18] J. Lee, W. Jeong, and B. Choi. A Scheduling Algorithm for Improving Scalability of LoRaWAN. In *International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1383–1388. IEEE, Oct 2018.

[LKM+16] Mads Lauridsen, István Z Kovács, Preben Mogensen, Mads Sorensen, and Steffen Holst. Coverage and capacity analysis of lte-m and nb-iot in a rural area. In *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, pages 1–5. IEEE, 2016.

[LLW+17] Xiaomin Li, Di Li, Jiafu Wan, Athanasios V Vasilakos, Chin-Feng Lai, and Shiyong Wang. A review of industrial wireless networks in the context of industry 4.0. *Wireless networks*, 23(1):23–41, 2017.

[LoR17a] LoRa Alliance, Technical Committee. LoRaWAN™ 1.1 Regional Parameters. `lora-alliance.org/sites/default/files/2018-04/lorawantm\_regional\_parameters\_v1.1rb\_-\_final.pdf`, 2017. Online; accessed 30-September-2019.

[LoR17b] LoRa Alliance, Technical Committee. LoRaWAN™ 1.1 Specification. `lora-alliance.org/sites/default/files/2018-04/lorawantm\_specification\_-v1.1.pdf`, 2017. Online; accessed 30-September-2019.

[LRK18] Shengyang Li, Usman Raza, and Aftab Khan. How agile is the adaptive data rate mechanism of lorawan? In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 206–212. IEEE, 2018.

[LSH08] Tomas Lennvall, Stefan Svensson, and Fredrik Hekland. A comparison of wirelesshart and zigbee for industrial applications. In *2008 IEEE International Workshop on Factory Communication Systems*, pages 85–88. IEEE, 2008.

[LYF20] Jiangbin Lyu, Dan Yu, and Liqun Fu. Achieving max-min throughput in lora networks. In *2020 International Conference on Computing, Networking and Communications (ICNC)*, pages 471–476. IEEE, 2020.

[MBCM19] Kais Mekki, Eddy Bajic, Frederic Chaxel, and Fernand Meyer. A comparative study of lpwan technologies for large-scale iot

deployment. *ICT express*, 5(1):1–7, 2019.

[MCV17] Davide Magrin, Marco Centenaro, and Lorenzo Vangelista. Performance Evaluation of LoRa Networks in a Smart City Scenario. In *ICC'17 SAC-7 IoT*, pages 1–7, May 2017.

[MHCK07] Gabriel Montenegro, Jonathan Hui, David Culler, and Nandakishore Kushalnagar. Transmission of IPv6 Packets over IEEE 802.15.4 Networks. RFC 4944, September 2007.

[Mis07] Ajay R Mishra. *Advanced cellular network planning and optimisation: 2G/2.5 G/3G... evolution to 4G*. John Wiley & Sons, 2007.

[MK20] Dinh Loc Mai and Myung Kyun Kim. Multi-hop lora network protocol with minimized latency. *Energies*, 13(6):1368, 2020.

[MKH+19] Bart Moons, Abdulkadir Karaagac, Jetmir Haxhibeqiri, Eli De Poorter, and Jeroen Hoebeke. Using schc for an optimized protocol stack in multimodal lpwan solutions. In *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, pages 430–435. IEEE, 2019.

[MLT08] R. Musaloiu-E., C. M. Liang, and A. Terzis. Koala: Ultra-Low Power Data Retrieval in Wireless Sensor Networks. In *2008 International Conference on Information Processing in Sensor Networks (ipsn 2008)*, pages 421–432, April 2008.

[MPH16] Konstantin Mikhaylov, Juha Petaejaejaervi, and Tuomo Haenninen. Analysis of capacity and scalability of the lora low power wide area network technology. In *European Wireless 2016; 22th European Wireless Conference*, pages 1–6. VDE, 2016.

[MPJ17] Konstantin Mikhaylov, Juha Petäjäjärvi, and Janne Janhunen. On LoRaWAN Scalability: Empirical Evaluation of Susceptibility to Inter-network Interference. In *Networks and Communications (EuCNC), 2017 European Conference on*, pages 1–6. IEEE, 2017.

[MTA20] Ana Minaburo, Laurent Toutain, and Ricardo Andreasen. LPWAN Static Context Header Compression (SCHC) for CoAP. Internet-Draft draft-ietf-lpwan-coap-static-context-hc-13, Internet Engineering Task Force, March 2020. Work in Progress.

[MTG+20] Ana Minaburo, Laurent Toutain, Carles Gomez, Dominique Barthel, and Juan-Carlos Zúñiga. SCHC: Generic Framework for Static Context Header Compression and Fragmentation. RFC 8724, April 2020.

[NCF+14] Thomas Nitsche, Carlos Cordeiro, Adriana B Flores, Edward W Knightly, Eldad Perahia, and Joerg C Widmer. Ieee 802.11 ad: directional 60 ghz communication for multi-gigabit-per-second wi-fi. *IEEE Communications Magazine*, 52(12):132–141, 2014.

[NFU20] Shusuke Narieda, Takeo Fujii, and Kenta Umebayashi. On power consumption of end device in lora networks with listen before talk. In *2020 International Conference on Information Networking (ICOIN)*, pages 297–300. IEEE, 2020.

[NH16] Kapileswar Nellore and Gerhard Hancke. A Survey on Urban Traffic Management System Using Wireless Sensor Networks. *Sensors*, 16(2):157, 2016.

[OG19] Behnam Ousat and Majid Ghaderi. Lora network planning: Gateway placement and device configuration. In *2019 IEEE International Congress on Internet of Things (ICIOT)*, pages 25–32. IEEE, 2019.

[OGS17] Rúben Oliveira, Lucas Guardalben, and Susana Sargento. Long Range Communications in Urban and Rural Environments. In *ISCC*, pages 810–817, July 2017.

[PAV+12] Maria Rita Palattella, Nicola Accettura, Xavier Vilajosana, Thomas Watteyne, Luigi Alfredo Grieco, Gennaro Boggia, and Mischa Dohler. Standardized protocol stack for the internet of (important) things. *IEEE communications surveys & tutorials*, 15(3):1389–1406, 2012.

[PC11] Stig Petersen and Simon Carlsen. Wirelesshart versus isa100. 11a: The format war hits the factory floor. *IEEE Industrial Electronics Magazine*, 5(4):23–34, 2011.

[PD08] Kris Pister and Lance Doherty. Tsmp: Time synchronized mesh protocol. *IASTED Distributed Sensor Networks*, 391:398, 2008.

[PDG+16] Maria Rita Palattella, Mischa Dohler, Alfredo Grieco, Gianluca Rizzo, Johan Torsner, Thomas Engel, and Latif Ladid. Inter-

net of things in the 5g era: Enablers, architecture, and business models. *IEEE Journal on Selected Areas in Communications*, 34(3):510–527, 2016.

[PGSDF20] Gopika Premsankar, Bissan Ghaddar, Mariusz Slabicki, and Mario Di Francesco. Optimal configuration of lora networks in smart cities. *IEEE Transactions on Industrial Informatics*, 2020.

[Pha18a] Congduc Pham. Investigating and experimenting csma channel access mechanisms for lora iot networks. In *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6. IEEE, 2018.

[Pha18b] Congduc Pham. Robust csma for long-range lora transmissions with image sensing devices. In *2018 Wireless Days (WD)*, pages 116–122. IEEE, 2018.

[PM20] Ladislav Polak and Jiri Milos. Performance analysis of lora in the 2.4 ghz ism band: coexistence issues with wi-fi. *Telecommunication Systems*, pages 1–11, 2020.

[PMHI16] Juha Petäjäjärvi, Konstantin Mikhaylov, Matti Hämäläinen, and Jari Iinatti. Evaluation of lora lpwan technology for remote health and wellbeing monitoring. In *2016 10th International Symposium on Medical Information and Communication Technology (ISMICT)*, pages 1–5. IEEE, 2016.

[PRKS17] Alexandru-Ioan Pop, Usman Raza, Parag Kulkarni, and Mahesh Sooriyabandara. Does Bidirectional Traffic Do More Harm Than Good in LoRaWAN based LPWA Networks? In *IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2017.

[PZ] Maciej Piechowiak and Piotr Zwierzykowski. Simulations of the mac layer in the lorawan networks.

[Res14] Machina Research. The need for low cost, high reach, wide area connectivity for the internet of things. a mobile network operator's prospective, 2014.

[RFF+17] M. Rizzi, P. Ferrari, A. Flammini, E. Sisinni, and M. Gidlund. Using lora for industrial wireless networks. In *IEEE 13th International Workshop on Factory Communication Systems (WFCS)*, pages 1–4, May 2017.

[RKS17] Usman Raza, Parag Kulkarni, and Mahesh Sooriyabandara. Low power wide area networks: An overview. *IEEE Communications Surveys & Tutorials*, 19(2):855–873, 2017.

[RMP16] B. Reynders, W. Meert, and S. Pollin. Range and Coexistence Analysis of Long Range Unlicensed Communication. In *23rd International Conference on Telecommunications (ICT)*, pages 1–6, May 2016.

[RMP17] B. Reynders, W. Meert, and S. Pollin. Power and spreading factor control in low power wide area networks. In *IEEE International Conference on Communications (ICC)*, pages 1–6, May 2017.

[RMZ+16] Rapeepat Ratasuk, Nitin Mangalvedhe, Yanji Zhang, Michel Robert, and Jussi-Pekka Koskinen. Overview of narrowband iot in lte rel-13. In *2016 IEEE conference on standards for communications and networking (CSCN)*, pages 1–7. IEEE, 2016.

[RWTP+18] Brecht Reynders, Qing Wang, Pere Tuset-Peiro, Xavier Vilajosana, and Sofie Pollin. Improving Reliability and Scalability of LoRaWANs through Lightweight Scheduling. *IEEE Internet of Things Journal*, 5(3):1830–1842, 2018.

[Sem15] Semtech Corporation. AN1200.22, LoRa™ Modulation Basics. `www.semtech.com/uploads/documents/an1200.22.pdf`, 2015. Online; accessed 30-September-2019.

[SGGMSI+20] Jesus Sanchez-Gomez, Jorge Gallego-Madrid, Ramon Sanchez-Iborra, Jose Santa, and Antonio F Skarmeta. Impact of schc compression and fragmentation in lpwan: A case study with lorawan. *Sensors*, 20(1):280, 2020.

[SGSGH19] Ruben M Sandoval, Antonio-Javier Garcia-Sanchez, and Joan Garcia-Haro. Optimizing and updating lora communication parameters: A machine learning approach. *IEEE Transactions on Network and Service Management*, 16(3):884–895, 2019.

[SHB14] Zach Shelby, Klaus Hartke, and Carsten Bormann. The constrained application protocol (coap). , 2014.

[She11] Asrar UH Sheikh. *Wireless communications: theory and techniques*. Springer Science & Business Media, 2011.

[SPL10] K Sandlund, G Pelletier, and Jonsson LE. The robust header compression (rohc) framework. Technical report, RFC 5795, March, 2010.

[SQN18] Binbin Su, Zhijin Qin, and Qiang Ni. Energy efficient resource allocation for uplink lora networks. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–7. IEEE, 2018.

[SRHGSGH20] Ruben M Sandoval, David Rodenas-Herraiz, Antonio-Javier Garcia-Sanchez, and Joan Garcia-Haro. Deriving and updating optimal transmission configurations for lora networks. *IEEE Access*, 8:38586–38595, 2020.

[SS05] Ananth Subramanian and Ali Sayed. Joint Rate and Power Control Algorithms for Wireless Networks. *IEEE Transactions on Signal Processing*, 53(11):4204–4214, Nov 2005.

[SVA18] Ioana Suciu, Xavier Vilajosana, and Ferran Adelantado. An analysis of packet fragmentation impact in lpwan. In *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6. IEEE, 2018.

[SYY+13] Zhengguo Sheng, Shusen Yang, Yifan Yu, Athanasios V Vasilakos, Julie A McCann, and Kin K Leung. A survey on the ietf protocol suite for the internet of things: Standards, challenges, and opportunities. *IEEE Wireless Communications*, 20(6):91–98, 2013.

[TD18a] Thanh-Hai To and Andrzej Duda. Simulation of lora in ns-3: Improving lora performance with csma. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE, 2018.

[TD18b] Thanh-Hai To and Andrzej Duda. Simulation of LoRa in NS-3: Improving LoRa Performance with CSMA. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE, 2018.

[TEG16] J. Toussaint, N. El Rachkidy, and A. Guitton. Performance Analysis of the On-the-air Activation in LoRaWAN. In *2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pages 1–7, Oct 2016.

[TH11] Pascal Thubert and Jonathan Hui. Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks. RFC 6282, September 2011.

[TKL+19a] Duc-Tuyen Ta, Kinda Khawam, Samer Lahoud, Cédric Adjih, and Steven Martin. Lora-mab: A flexible simulator for decentralized learning resource allocation in iot networks. In *2019 12th IFIP Wireless and Mobile Networking Conference (WMNC)*, pages 55–62. IEEE, 2019.

[TKL+19b] Duc-Tuyen Ta, Kinda Khawam, Samer Lahoud, Cédric Adjih, and Steven Martin. Lora-mab: Toward an intelligent resource allocation approach for lorawan. In *GLOBECOM 2019-IEEE Global Communications Conference*, 2019.

[TXZ05] Jian Tang, Guoliang Xue, and Weiyi Zhang. Interference-aware topology control and qos routing in multi-channel wireless mesh networks. In *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 68–77, 2005.

[Van17] Lorenzo Vangelista. Frequency shift chirp modulation: The lora modulation. *IEEE Signal Processing Letters*, 24(12):1818–1821, 2017.

[VGZK16] Mikhail Vilgelm, Murat Gürsu, Samuele Zoppi, and Wolfgang Kellerer. Time slotted channel hopping for smart metering: Measurements and analysis of medium access. In *2016 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 109–115. IEEE, 2016.

[VMP19] Rafael Vidal, Nicolas Montavont, and Georgios Z Papadopoulos. Lorawan schc fragmentation demystified. In *Ad-Hoc, Mobile, and Wireless Networks: 18th International Conference on Ad-Hoc Networks and Wireless, ADHOC-NOW 2019, Luxembourg, Luxembourg, October 1–3, 2019, Proceedings*, volume 11803, page 213. Springer Nature, 2019.

[vS13] Stephen Brown and Cormac Sreenan. Software updating in wireless sensor networks: A survey and lacunae. *Journal of Sensor and Actuator Networks*, 2(4):717–760, 2013.

[VS17] N. Varsier and J. Schwoerer. Capacity Limits of LoRaWAN Technology for Smart Metering Applications. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6, May 2017.

[WALJ+06] Geoff Werner-Allen, Konrad Lorincz, Jeff Johnson, Jonathan Lees, and Matt Welsh. Fidelity and Yield in a Avolcano Monitoring Sensor Network. In *Proceedings of the 7th symposium on Operating systems design and implementation*, pages 381–396. USENIX Association, 2006.

[WBII06] Sonia Waharte, Raouf Boutaba, Youssef Iraqi, and Brent Ishibashi. Routing protocols in wireless mesh networks: challenges and design considerations. *Multimedia tools and Applications*, 29(3):285–303, 2006.

[Win62] MR Winkler. Chirp signals for communication. *IEEE WESCON Convention Record*, 1962.

[WK03] C. Westphal and R. Koodli. Ip header compression: a study of context establishment. In *2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003.*, volume 2, pages 1025–1031 vol.2, March 2003.

[WKL+16] Andrew J Wixted, Peter Kinnaird, Hadi Larijani, Alan Tait, Ali Ahmadinia, and Niall Strachan. Evaluation of lora and lorawan for wireless sensor networks. In *2016 IEEE SENSORS*, pages 1–3. IEEE, 2016.

[WLA+17] Y-P Eric Wang, Xingqin Lin, Ansuman Adhikary, Asbjorn Grovlen, Yutao Sui, Yufei Blankenship, Johan Bergman, and Hazhir S Razaghi. A primer on 3gpp narrowband internet of things. *IEEE Communications Magazine*, 55(3):117–123, 2017.

[WPG15] Thomas Watteyne, Maria-Rita Palattella, and Luigi Alfredo Grieco. Using ieee 802.15. 4e time-slotted channel hopping (tsch) in the internet of things (iot): Problem statement. *URL https://tools. ietf. org/html/draft-ietf-6tisch-tsch-05*, 2015.

[WWDS15] Thomas Watteyne, Joy Weiss, Lance Doherty, and Jonathan Simon. Industrial ieee802. 15.4 e networks: Performance and

trade-offs. In *2015 IEEE International Conference on Communications (ICC)*, pages 604–609. IEEE, 2015.

[Yat95a] Roy Yates. A Framework for Uplink Power Control in Cellular Radio Systems. *IEEE Journal on Selected Areas in Communications*, 13(7):1341–1347, Sep 1995.

[Yat95b] Roy D. Yates. A framework for uplink power control in cellular radio systems. *IEEE Journal on selected areas in communications*, 13(7):1341–1347, 1995.

[YXL12] Bin Yu, Lisheng Xu, and Yongxu Li. Bluetooth low energy (ble) based mobile electrocardiogram monitoring system. In *2012 IEEE International Conference on Information and Automation*, pages 763–767. IEEE, 2012.

[YYW+05] Yuan Yuan, Hao Yang, Starsky HY Wong, Songwu Lu, and William Arbaugh. Romer: Resilient opportunistic mesh routing for wireless mesh networks. In *IEEE workshop on wireless mesh networks (WiMesh)*, volume 12, 2005.

[ZAC+19a] Dimitrios Zorbas, Khaled Q. Abdelfadeel, Victor Cionca, Dirk Pesch, and Brendan O'Flynn. Offline scheduling algorithms for Time-slotted LoRa-based Bulk Data Transmission. In *IEEE 5th World Forum on Internet of Things (WFIoT)*, pages 1–6. IEEE, 2019.

[ZAC+19b] Dimitrios Zorbas, Khaled Q Abdelfadeel, Victor Cionca, Dirk Pesch, and Brendan O'Flynn. Offline scheduling algorithms for time-slotted lora-based bulk data transmission. In *the IEEE 5th World Forum on Internet of Things (WF-IoT). IEEE*, pages 1–6, 2019.

[ZHR+18] Torsten Zimmermann, Jens Hiller, Helge Reelfs, Pascal Hein, and Klaus Wehrle. Split: Smart protocol loading for the iot. In *EWSN*, pages 49–54, 2018.

[ZO18] Dimitrios Zorbas and Brendan O'Flynn. Collision-Free Sensor Data Collection using LoRaWAN and Drones. In *Global Information Infrastructure and Networking Symposium (GIIS)*. IEEE, Oct 2018.

[ZSA+19] KOEN Zandberg, KASPAR Schleiser, FRANCISCO Acosta, HANNES Tschofenig, and EMMANUEL Baccelli. Secure firmware updates for constrained iot devices using open standards: A reality check. *IEEE Access*, 2019.

[ZSH+19] Tao Zhou, Yuyi Sun, Shibo He, Zhiguo Shi, Jiming Chen, and Zhen Tao. Gateway planning for hybrid lora networks. In *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 1071–1079. IEEE, 2019.